Tektronix®
COMMITTED TO EXCELLENCE

PLEASE CHECK FOR CHANGE INFORMATION AT THE REAR OF THIS MANUAL.

4052R11

CHARACTER AND SYMBOL ROM PACK

INSTRUCTION MANUAL

Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97077

MANUAL PART NO.
070-3794-00

First Printing MAR 1981

WARNING

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual, may cause interference to radio communications. It has been tested and found to comply with the limits for Class A computing devices pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the users at their own expense will be required to take whatever measures may be required to correct the interference.

MANUAL REVISION STATUS

PRODUCT: 4052R11 Character and Symbol ROM Pack

This manual supports the following versions of this product: Serial Numbers B010100 and up.

REV.	DATE	DESCRIPTION
@	MAR 1981	Original Issue

CONTENTS

Section 1	**GENERAL DESCRIPTION**	**Page**
	Introduction	1-1
	Configuration	1-1
	Service Information	1-2
	Specifications	1-2
	Environmental Characteristics	1-2
	Physical Characteristics	1-3
	Standard Accessories	1-3
	Installing the ROM Pack	1-4
Section 2	**ROUTINE DESCRIPTIONS**	
	Introduction	2-1
	Character Transforms	2-2
	Character Size and Spacing	2-3
	The LSIZE Routine	2-5
	The LDIM Routine	2-6
	The LRATIO Routine	2-7
	The LSCALE Routine	2-8
	Character Rotation and Slanting	2-9
	The LROT Routine	2-10
	The LSLANT Routine	2-11
	Character Fonts	2-12
	The LMFONT Routine	2-17
	The LAFONT Routine	2-18
	Character Resolution	2-20
	The LQUAL Routine	2-21
	The LSMOTH Routine	2-22
	Cursor Position	2-23
	The LMOVE Routine	2-24
	The LHOME Routine	2-25
	Character Display	2-26
	The LETTER Routine	2-27
	The LCENTR Routine	2-29
	The LVERTI Routine	2-31
	The LETEVN Routine	2-33
	The LRESET Routine	2-34

Section 2 (cont) **Page**

Arcs and Circles 2-35
The ARC1 Routine 2-36
The ARC2 Routine 2-38
The CIRCLE Routine 2-40
The SMOOTH Routine 2-41
Character and Symbol Generation 2-42
The STROKE Routine 2-42
Enhancements to Graphics 2-45
The TYPEKY Routine 2-46
The CURSOR Routine 2-47
The ENDRAG Routine 2-48
The CLDRAG Routine 2-50

Section 3 **REPLACEABLE PARTS**

Section 4 **DIAGRAMS**

Appendix A **VERIFICATION PROGRAM**

Appendix B **COMMAND SUMMARY**

Appendix C **UNDERSTANDING ERRORS**

INDEX

ILLUSTRATIONS

Figure	Description	Page
1-1	4052R11 Character and Symbol ROM Pack	v
1-2	Installing the ROM Pack in the Graphic System	1-4
2-1	Character Cells and Character Envelopes	2-3
2-2	Setting Character Dimensions	2-4
2-3	The LSIZE Routine with Negative Arguments	2-5
2-4	Rotation with the LROT Routine	2-9
2-5	Slanting with the LSLANT Routine	2-9
2-6	Fonts 0 — 9 (4052/4054 Character Fonts)	2-13
2-7	Font 10	2-14
2-8	Font 11	2-15
2-9	Font 12	2-16
2-10	The LCENTR Routine Example	2-29
2-11	The LVERTI Routine Example	2-31
2-12	The LETEVN Routine Example	2-33
2-13	How To Draw an Ellipse	2-35
2-14	The ARC1 Routine	2-36
2-15	The ARC2 Routine	2-38
2-16	The CIRCLE Routine Example	2-40
2-17	The STROKE Routine Example	2-42
A-1	Output From the Verification Program	A-3

TABLES

Table	Description	Page
2-1	Initial Routine Settings	2-34

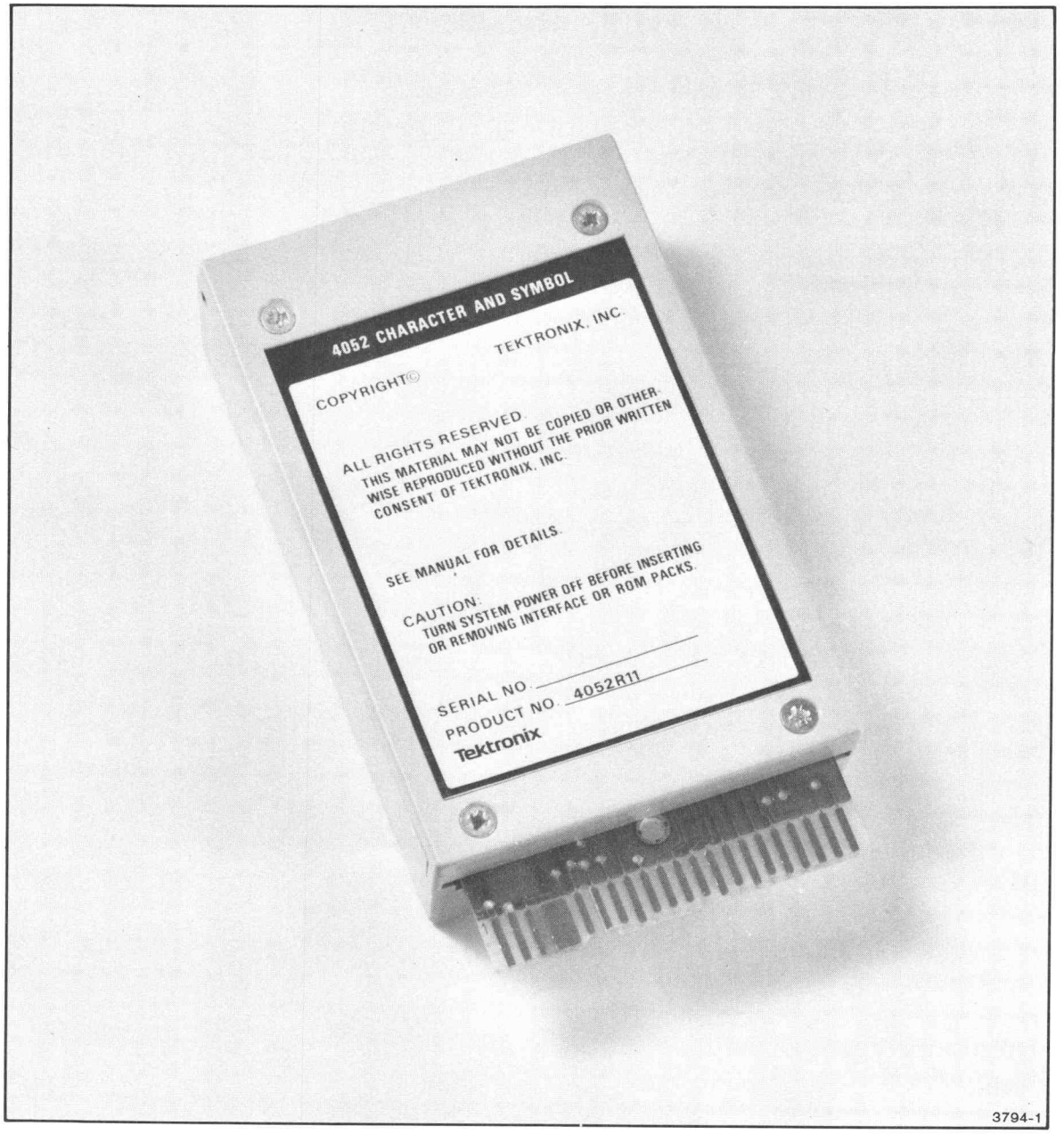


Figure 1-1. 4052R11 Character and Symbol ROM Pack.

Section 1

GENERAL DESCRIPTION

INTRODUCTION

The 4052R11 Character and Symbol ROM Pack provides additional and enhanced character display features to the TEKTRONIX 4052 and 4054 Graphic Computing Systems. Major features in this ROM pack include:

- Thirteen enhanced alphanumeric character fonts. Ten of these fonts are identical to the fonts already in the 4052 and 4054 Graphic Computing Systems and are included in this ROM pack so that they may be used with the other special display features of the Character and Symbol ROM Pack. Three new character fonts are provided with the Character and Symbol ROM Pack. These fonts provide special math, engineering, and drafting symbols.

- Capability for continuously variable proportioning, rotating, scaling, and slanting of characters or character strings. The characters may come from any of the thirteen character fonts, or from a combination of fonts.

- Increased character quality and smoothness.

- Capability to design custom symbols using X,Y coordinates. These symbols might be special figures, characters, or company logos. These symbols may also be scaled, rotated, and slanted.

- Fast, easy generation of arcs and circles of specified smoothness.

- Additional graphics functions, adding to the capabilities of the POINTER statement and 4054 Option 30, Dynamic Graphics.

CONFIGURATION

The 4052R11 Character and Symbol ROM Pack may be installed in any backpack slot of a 4052 or 4054 Graphic Computing System. This ROM pack also includes special features that only operate with 4054 Option 30. The 4052R11 Character and Symbol ROM Pack is not compatible with the 4051 Graphic Computing System.

The 4052 Graphic System must have firmware version 4.3 or higher installed.

SERVICE INFORMATION

This manual contains parts lists and schematics for the ROM pack but does not contain any service procedures. Refer to the 4052/4054 Technical Data Service Manual for a general description of ROM pack circuitry. This ROM pack requires no adjustments or preventive maintenance.

SPECIFICATIONS

The terms "performance requirement" and "supplemental information" are used as follows to describe the ROM pack's specifications:

Performance Requirement: a statement that defines a characteristic in quantitative terms of performance, usually in limit form.

Supplemental Information: statements that amplify or supplement performance requirements or that provide performance information.

Environmental Characteristics

The 4052R11 Character and Symbol ROM Pack meets the environmental specifications of TEKTRONIX 4052/4054 Series Graphic Computing Systems.

Characteristics	Performance Requirements
Temperature	
Operating	+ 50 to + 104° F (+ 10 to + 40° C)
Non-operating	−40 to + 149° F (−40 to + 65° C)
Altitude	
Operating	15,000 ft maximum (4,572 m)
Non-operating	50,000 ft maximum (15,240 m)
Humidity	
Operating	80% non-condensing
Storage	95% non-condensing

Physical Characteristics

Characteristics	Supplemental Information
Dimensions[1]	
Length	4.7 in (11.84 cm)
Width	2.620 in (6.65 cm)
Depth	0.875 in (2.22 cm)

STANDARD ACCESSORIES

4052R11 Character and Symbol ROM Pack Instruction Manual

[1] Including edge-board connector.

INSTALLING THE ROM PACK

The 4052R11 Character and Symbol ROM Pack can be installed in any backpack slot of a TEKTRONIX 4052 or 4054 Graphic Computing System. The 4052R11 is not compatible with the TEKTRONIX 4051.

1. Be sure the power switch is off.

CAUTION

Inserting the ROM pack while the power is ON may cause damage to the ROM pack. The Graphic System memory contents may also be lost. Turn the Graphic System's power OFF before inserting the ROM pack.

2. Insert the Character and Symbol ROM Pack into one of the slots in the rear of the Graphic System as shown in Figure 1-2. Press down and gently rock the plastic case until the ROM pack edge-board connector is firmly seated in the receptacle.

3. Turn the power switch ON and wait a few seconds for the system to warm up.

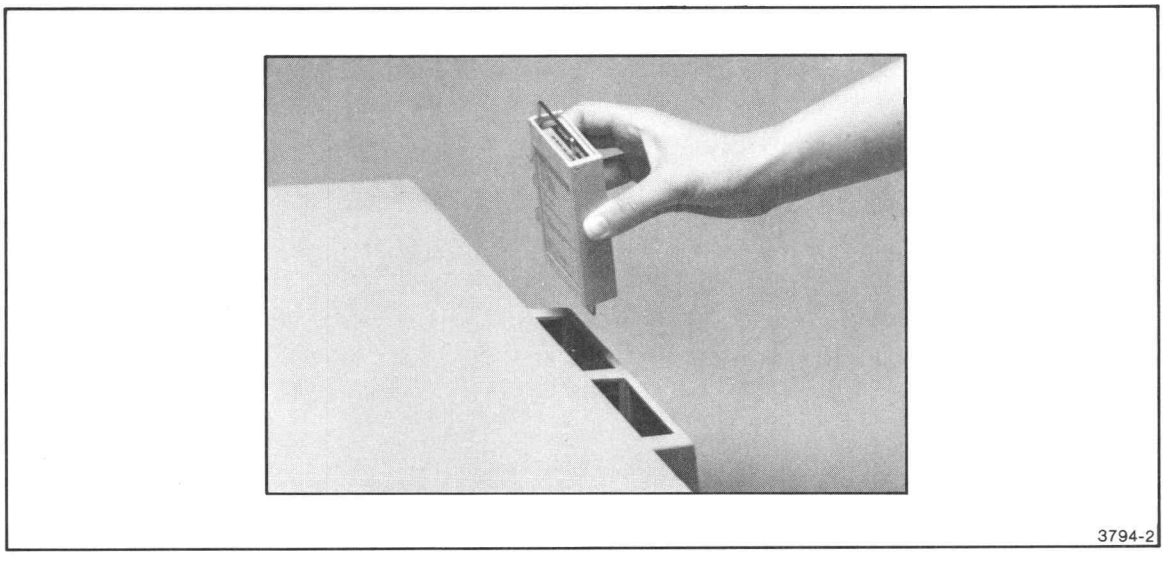


Figure 1-2. Installing the ROM Pack in the Graphic System.

Section 2

ROUTINE DESCRIPTIONS

INTRODUCTION

The 4052R11 Character and Symbol ROM Pack adds twenty-six new routines into the TEKTRONIX 4052 or 4054 Graphic Computing Systems. These routines serve four distinct types of functions. They can be classified as:

- Routines which affect the display of alphanumeric characters. These functions are described under Character Transforms.
- Routines which draw smoothed arcs and circles. These routines are described under Arcs and Circles.
- Routines which generate special custom symbols. This routine is described under Character and Symbol Generation.
- Routines which provide additional graphics enhancements, especially when used with 4054 Option 30. These routines are described under Graphics Enhancement.

This section contains a detailed description of each of these twenty-six call routines. The explanations are intended for users who are familiar with the operation of TEKTRONIX 4052 and 4054 Graphic Computing Systems. The explanations include basic information about the use of the routines. Examples are included to help clarify the use of the ROM pack.

The ROM pack routines are accessed using the CALL statement in 4050 Series BASIC. The routine name is specified in the CALL statement. Refer to the 4050 Series Graphic System Reference Manual for more information on CALL statements.

CHARACTER TRANSFORMS

The Character Transform routines can be further grouped into routines which:

- Adjust character size and spacing;
- Rotate and slant characters and lines;
- Choose different character fonts;
- Improve character quality by giving greater resolution;
- Position the cursor;
- Display characters horizontally, vertically, centered on a point, or with proportionately adjusted spacing;
- Reset all character routine settings without affecting other settings.

These routines are all interconnected. Some routines set various parameters which are used by the display routines in this group. All Character Transform routine names also start with the letter "L" (for "letter") because these routines affect how letters or characters appear.

As an example, size and slant are two parameters which may be set by routines in this ROM pack. In order to print a character or string with the adjusted size and slant, the special display routines provided in the ROM pack must be used. A normal PRINT statement ignores these parameters and prints the characters as if the ROM pack is not there.

Character Size and Spacing

Figure 2-1 shows the character cell used in the 4052 and 4054 Graphic Computing Systems. The Character and Symbol ROM Pack uses character envelopes. The character envelope, like the character cell, is the space within which a character is displayed. As Figure 2-1 shows, a character is positioned differently within the ROM pack routines. The position of a character within the envelope has bearing on how the ROM pack routines display characters.

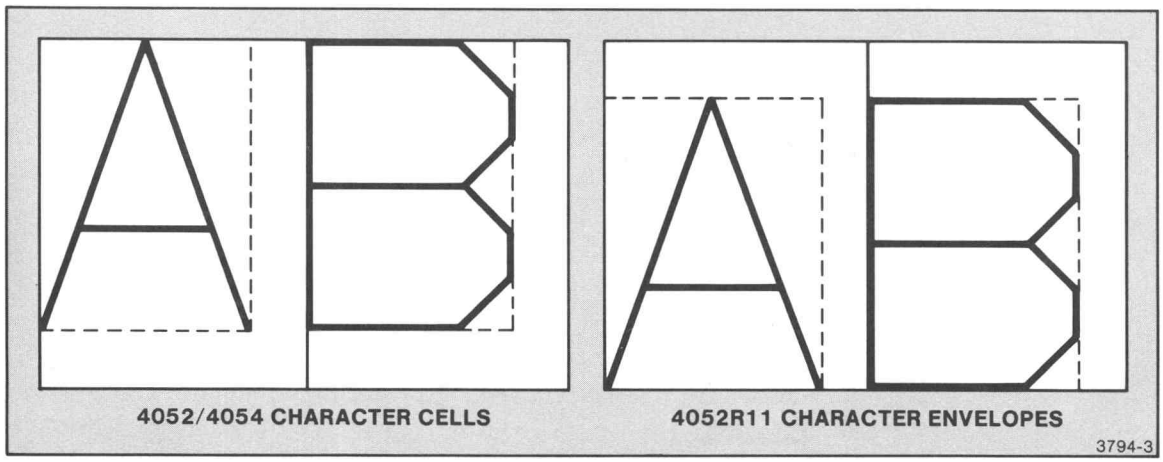


Figure 2-1. Character Cells and Character Envelopes.

Four routines may be used to adjust character size and spacing:

- LSIZE may be used to specify relative character or letter size.
- LDIM may be used to specify relative character or letter envelope size.
- LRATIO may be used to specify the ratio between character size and envelope size.
- LSCALE may be used to proportionately adjust both the character size and the envelope size.

Figure 2-2 illustrates the dimensions set by these routines.

Only two of the first three routines are necessary to adjust both character and envelope size. If both character size and spacing (envelope size) are to be adjusted, either LSIZE and LDIM, LDIM and LRATIO, or LSIZE and LRATIO may be used.

If only one routine is called, default values will be used for character size or spacing. The letter size default has precedence. For example, if LRATIO is the only routine called, the normal character size is used, and the envelope size is adjusted appropriately.

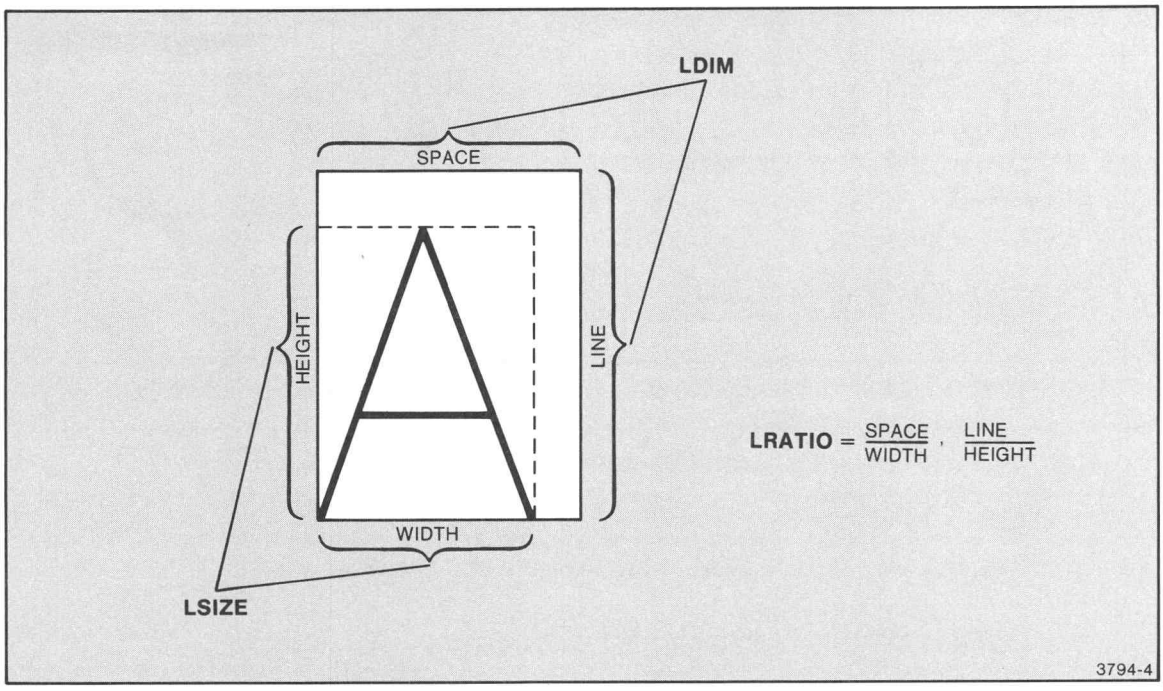


Figure 2-2. Setting Character Dimensions.

The LSIZE Routine

Syntax Form:

<line number> CAL "LSIZE",numeric expression,numeric expression

Descriptive Form:

<line number> CALL "LSIZE",width, height

Purpose

The LSIZE routine sets the width and height of average characters. The width and height must be expressed in "user data units," as specified by the current WINDOW.

Examples

```
200 WINDOW 0,10,0,10
210 CALL "LSIZE",1,2
```

This example sets a size for any characters subsequently printed with one of the Character Display routines. In this case, characters are proportioned such that an average uppercase character is one-tenth of the window size wide and one-fifth of the window size tall.

Explanation

The numeric expressions in the LSIZE routine may be any real numbers. Negative values cause reflection around the x and y axes.

The LSIZE routine sets the character size of average uppercase characters. Some characters will be narrower (I or 1, for example). Lowercase characters are proportioned accordingly.

The default LSIZE calls for the normal character size when using a 4052, or character size 4, the normal default, when used on a 4054. This is 1.14 by 1.79 user data units.

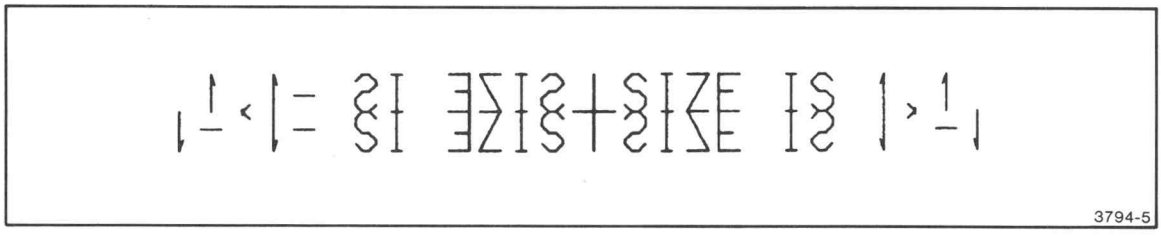


Figure 2-3. The LSIZE Routine with Negative Arguments.

The LDIM Routine

Syntax Form:

<line number> CAL "LDIM",numeric expression,numeric expression

Descriptive Form:

<line number> CALL "LDIM",space width,line height

Purpose

The LDIM routine sets the width and height of spaces and lines. The width and height must be expressed in user data units, as specified by the current WINDOW.

Examples

```
200 WINDOW 0,10,0,10
220 CALL "LDIM",1,2
```

This example sets the spacing for any characters subsequently printed with one of the Character Display routines. Ten characters can fit on one line, and five lines can show in the window.

Explanation

While LSIZE sets the actual character width and height, LDIM sets the distance between the start of one character and the start of the next. This distance is the character width plus the space between characters, and the character height plus the space between lines.

LDIM is calculated from expressions for the LSIZE and LRATIO routines. The default character envelope size is 1.71 by 2.29 user data units.

As with LSIZE, the numeric expressions may be any real numbers. Negative values for LDIM also cause reflection around the x and y axes.

The LRATIO Routine

Syntax Form:

<line number> CAL "LRATIO",numeric expression,numeric expression

Descriptive Form:

<line number> CALL "LRATIO", width ratio, height ratio

Purpose

The LRATIO routine sets the ratios between character width and spacing between characters, and between character height and spacing between lines.

Examples

```
200 WINDOW 0,10,0,10
210 CALL "LSIZE",1,2
220 CALL "LRATIO",1.2,1.2
```

or

```
200 WINDOW 0,10,0,10
210 CALL "LDIM",1.2,2.4
220 CALL "LRATIO",1.2,1.2
```

These two examples establish the same character and envelope size. The first uses the LRATIO routine to extract the envelope size. The second extrapolates the character size from LRATIO and the envelope size.

Explanation

The LRATIO routine has the form:

$$\text{CALL "LRATIO"}, \quad \frac{\text{character spacing}}{\text{character width}} \quad , \quad \frac{\text{line height}}{\text{character height}}$$

Values of the numeric expressions for LRATIO may be any real numbers. Values less than one cause character overlap.

Negative values cause characters to be displayed in reverse order or upside down with reversed line feeds.

The default LRATIO setting is (1.5,1.64).

The LSCALE Routine

Syntax Form:

<line number> CAL "LSCALE",numeric expression,numeric expression

Descriptive Form:

<line number> CALL "LSCALE",x-axis factor,y-axis factor

Purpose

The LSCALE routine proportionately magnifies or reduces character and space size.

Examples

```
200 WINDOW 0,10,0,10
210 CALL "LSIZE",1,2
220 CALL "LDIM",1.2,2.2

300 CALL "LSCALE",0.8,1.5
```

The LSCALE routine in line 300 has the effect of changing both LSIZE and LDIM. The effective LSIZE becomes (.8,3); the effective LDIM size becomes (.96,3.3).

Explanation

The LSCALE routine maintains the set character size/envelope size ratio. It allows the character size and envelope size to be magnified or diminished in one statement, rather than the two lines required to set the proportions.

The initial or default LSCALE setting is (1,1).

Character Rotation and Slanting

The Character and Symbol ROM Pack includes two routines to specify the rotation and slanting of characters and character strings.

- LROT causes subsequent Character Display routines to rotate the entire print line to a specified angle.
- LSLANT causes subsequent Character Display routines to slant each character to a specified angle.

In order to use these routines most effectively, remember:

- The origin for both rotation and slanting is the lower left corner of the character envelope. This one starting point is not affected by rotation or slant.
- The angle of rotation or slant depends on the environmental setting. The angle units may be set as degrees, radians, or grads.

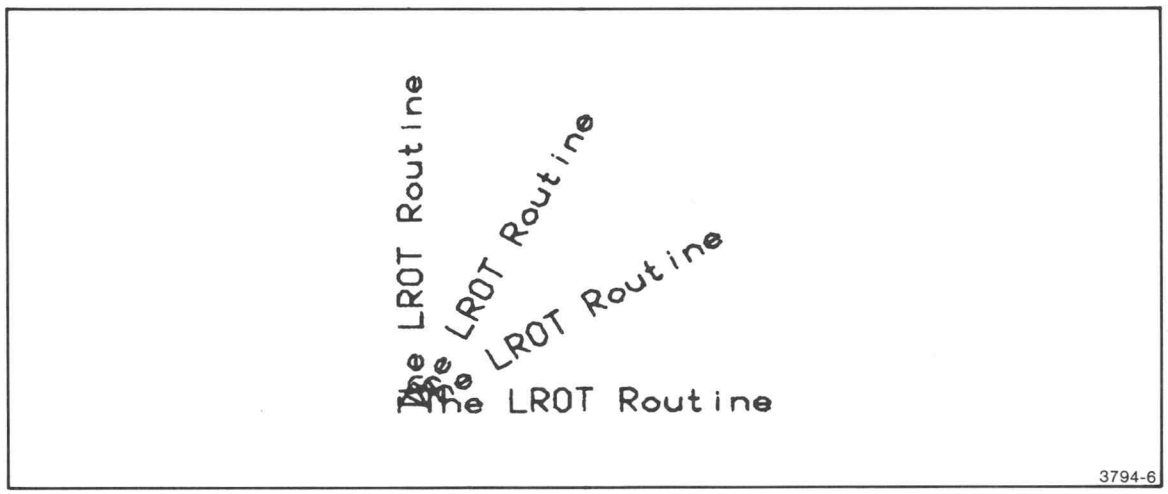


Figure 2-4. Rotation with the LROT Routine.

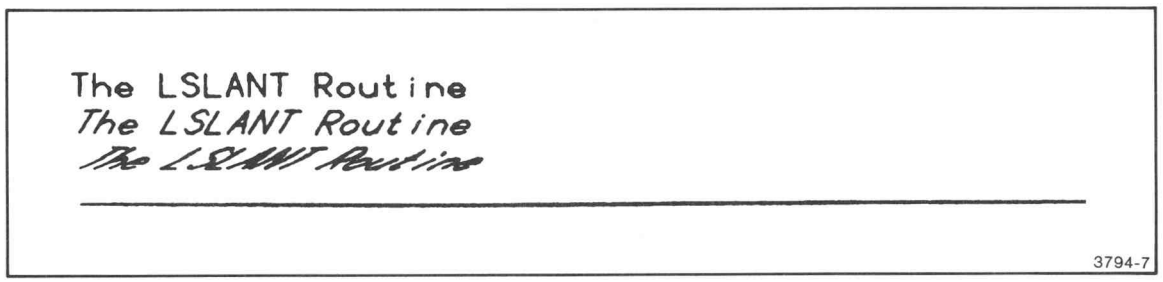


Figure 2-5. Slanting with the LSLANT Routine.

The LROT Routine

Syntax Form:

<line number> CAL "LROT",numeric expression

Descriptive Form:

<line number> CALL "LROT",angle

Purpose

The LROT routine causes a character or character string to be rotated at a specified angle when printed by a Character Display routine.

Examples

```
300 SET DEGREES
310 CALL "LROT",30
```

This example causes any character or character string when called by a Character Display routine to be displayed on a 30° angle (counterclockwise from the horizontal x-axis). The 30° rotation is maintained until the LROT routine is changed or reset (see the LRESET function).

Explanation

The rotation angle may be specified by radians, degrees, or grads. (Remember that the default is radians.) Positive angles are always defined counterclockwise from the horizontal. Negative angles are defined clockwise from horizontal. The origin is the lower left of the starting space or character (except when using the LCENTER routine).

The default LROT value is zero; no rotation.

The LSLANT Routine

Syntax Form:

<line number> CAL "LSLANT",numeric expression

Descriptive Form:

<line number> CALL "LSLANT",angle

Purpose

The LSLANT routine causes any characters called by a Character Display routine to be slanted at a specified angle.

Examples

```
300 SET DEGREES
320 CALL "LSLANT",-15
```

This example causes any characters subsequently called by a Character Display routine to slant 15° to the left (counterclockwise).

Explanation

LSLANT specifies the angle in radians, degrees, or grads, whichever has been SET. The angle is measured from the vertical y-axis. Positive values slant characters clockwise to the right; negative values slant counterclockwise, to the left.

Angles are permitted within the range of −90° to 90° or the equivalent angle in radians or grads. Angles greater than 89.9° or the equivalent are rounded to 89.9°. Angles of −89.9° or less are rounded to −89.9°. Characters can not be slanted below the horizontal x-axis. Characters may be slanted and then reflected (with a negative LSIZE, for example).

The default LSLANT is zero; no slant.

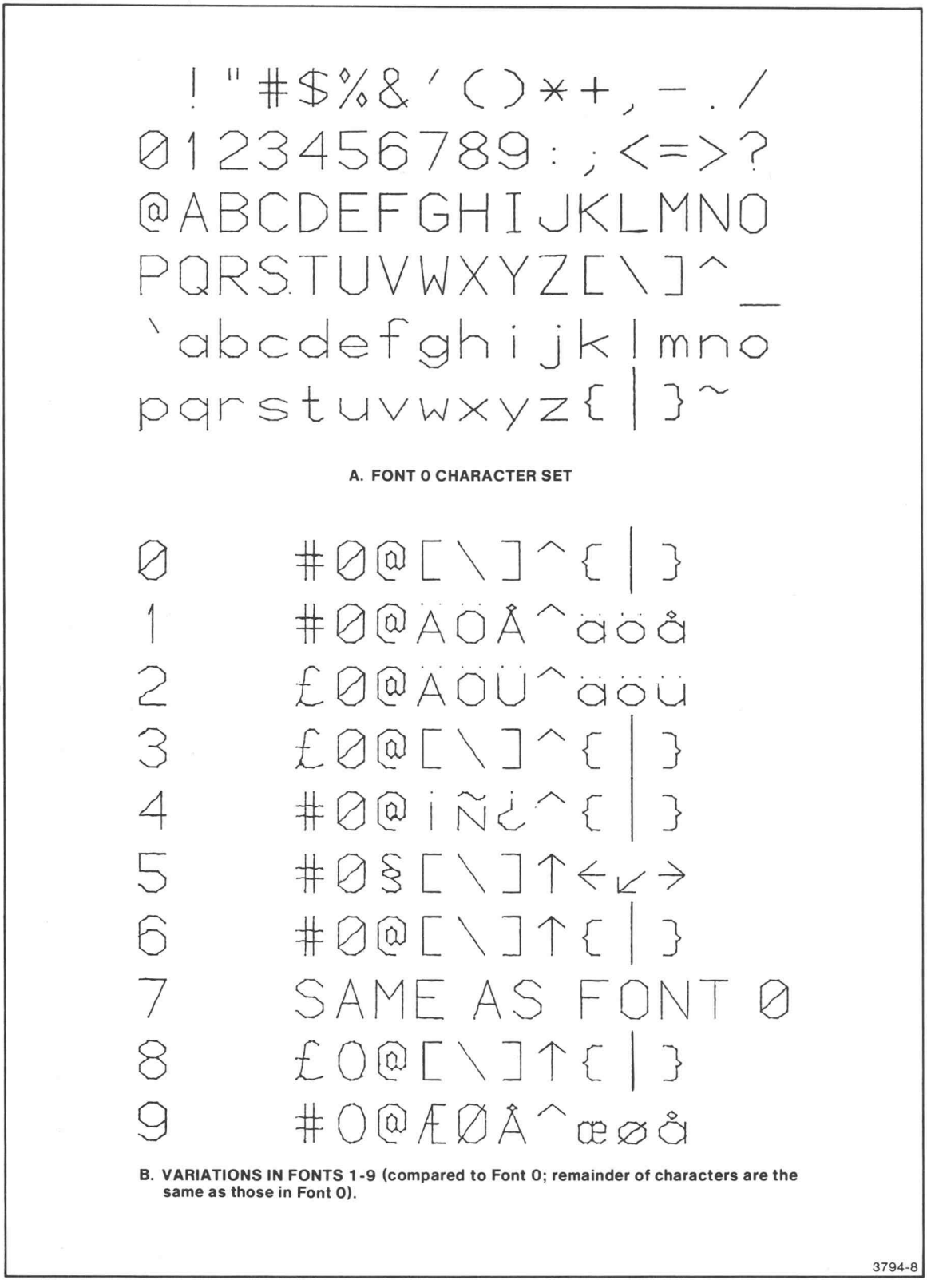
Character Fonts

The Character and Symbol ROM Pack includes two routines to select character fonts.

- LMFONT selects the main character font. Character Display routines normally use this font.

- LAFONT selects the alternate character font. Control characters included within a print string dictate which font is used for displayed characters. This allows characters from two different character fonts to be included in the same character string. (Refer to the LAFONT discussion.)

Thirteen character fonts are provided in the Character and Symbol ROM Pack. The first ten (0 through 9) are identical to the existing 4052/4054 character fonts. They are included so that they may be used with the Character Transform routines in this ROM pack.



A. FONT 0 CHARACTER SET

B. VARIATIONS IN FONTS 1-9 (compared to Font 0; remainder of characters are the same as those in Font 0).

3794-8

Figure 2-6. Fonts 0 — 9 (4052/4054 Character Fonts).

	SP 32	Ø	Ø 48	@	@ 64	Π	P 80	\	\ 96	π	p 112
!	! 33	1	1 49	A	A 65	Q	Q 81	α	a 97	q	q 113
"	" 34	2	2 50	B	B 66	P	R 82	β	b 98	ρ	r 114
#	# 35	3	3 51	Ψ	C 67	Σ	S 83	ψ	c 99	σ	s 115
$	$ 36	4	4 52	Δ	D 68	T	T 84	δ	d 100	τ	t 116
%	% 37	5	5 53	E	E 69	Θ	U 85	ε	e 101	ϑ	u 117
&	& 38	6	6 54	Φ	F 70	Ω	V 86	φ	f 102	ω	v 118
'	' 39	7	7 55	Γ	G 71	W	W 87	γ	g 103	ς	w 119
(	(40	8	8 56	H	H 72	X	X 88	η	h 104	χ	x 120
)	) 41	9	9 57	I	I 73	ϒ	Y 89	ι	i 105	υ	y 121
*	* 42	:	: 58	Ξ	J 74	Z	Z 90	ξ	j 106	ζ	z 122
+	+ 43	;	; 59	K	K 75	[	[91	κ	k 107	{	{ 123
,	, 44	<	< 60	Λ	L 76	\	\ 92	λ	l 108	∂	| 124
—	- 45	=	= 61	M	M 77	]	] 93	μ	m 109	}	} 125
.	. 46	>	> 62	N	N 78	^	^ 94	ν	n 110	~	~ 126
/	/ 47	?	? 63	O	O 79	_	_ 95	o	o 111		DEL 127

3794-9

Figure 2-7. Font 10.

	SP 32	0	0 48	2	@ 64	∏	P 80	-1	` 96	π	p 112
·	! 33	∫	1 49	A	A 65	√	Q 81	α	a 97	—	q 113
□	" 34	∴	2 50	B	B 66	P	R 82	β	b 98	ρ	r 114
○	# 35	∃	3 51	ψ	C 67	Σ	S 83	ψ	c 99	σ	s 115
△	$ 36	⊂	4 52	Δ	D 68	T	T 84	δ	d 100	τ	t 116
+	% 37	∪	5 53	E	E 69	Θ	U 85	ε	e 101	ϑ	u 117
◇	& 38	⊃	6 54	ϕ	F 70	Ω	V 86	φ	f 102	ω	v 118
⊠	' 39	∩	7 55	Γ	G 71	W	W 87	γ	g 103	ς	w 119
⌑	(40	∞	8 56	H	H 72	X	X 88	η	h 104	χ	x 120
⊕	) 41	▽	9 57	I	I 73	ϒ	Y 89	ι	i 105	υ	y 121
⊞	* 42	:	: 58	Ξ	J 74	Z	Z 90	ξ	j 106	ζ	z 122
+	+ 43	≠	; 59	K	K 75	[	[91	κ	k 107	ℓ	{ 123
⊥	, 44	≤	< 60	Λ	L 76	±	\ 92	λ	l 108	∂	| 124
‖	- 45	≡	= 61	M	M 77	]	] 93	μ	m 109	↓	} 125
.	. 46	≥	> 62	N	N 78	↑	^ 94	ν	n 110	~	~ 126
÷	/ 47	∝	? 63	O	O 79	_	_ 95	o	o 111		DEL 127

3794-10

Figure 2-8. Font 11.

	SP 32	0	0 48	@	@ 64	P	P 80	\	\ 96	p	p 112
!	! 33	1	1 49	A	A 65	Q	Q 81	a	a 97	q	q 113
"	" 34	2	2 50	B	B 66	R	R 82	b	b 98	r	r 114
#	# 35	3	3 51	C	C 67	S	S 83	c	c 99	s	s 115
$	$ 36	4	4 52	D	D 68	T	T 84	d	d 100	t	t 116
%	% 37	5	5 53	E	E 69	U	U 85	e	e 101	u	u 117
&	& 38	6	6 54	F	F 70	V	V 86	f	f 102	v	v 118
'	' 39	7	7 55	G	G 71	W	W 87	g	g 103	w	w 119
(	(40	8	8 56	H	H 72	X	X 88	h	h 104	x	x 120
)	) 41	9	9 57	I	I 73	Y	Y 89	i	i 105	y	y 121
*	* 42	:	: 58	J	J 74	Z	Z 90	j	j 106	z	z 122
+	+ 43	;	; 59	K	K 75	μ	[91	k	k 107	Ω	{ 123
,	, 44	<	< 60	L	L 76	°	\ 92	l	l 108	±	| 124
–	– 45	=	= 61	M	M 77	ø	] 93	m	m 109	¢	} 125
.	. 46	>	> 62	N	N 78	^	^ 94	n	n 110	~	~ 126
/	/ 47	?	? 63	O	O 79	_	_ 95	o	o 111		DEL 127

3794-11

Figure 2-9. Font 12.

The LMFONT Routine

Syntax Form:

<line number> CAL "LMFONT",numeric expression

Descriptive Form:

<line number> CALL "LMFONT",font number

Purpose

The LMFONT routine selects one of the thirteen available print fonts.

Examples

```
400 CALL "LMFONT",11
```

Explanation

The font number in this routine may be any real number. Non-integer values are rounded to the nearest integer, 0 through 12. Values greater than twelve or less than zero default to Font 0. Thirteen character fonts, Fonts 0 through 12, are available in the Character and Symbol ROM Pack. All characters displayed by a Character Display routine are normally chosen from the font specified by the LMFONT routine. If an LMFONT is not specified, Font 0 is used. Font 0 is the normal 4052/4054 ASCII character font.

The LAFONT Routine

Syntax Form:

<line number> CAL "LAFONT",numeric expression

Descriptive Form:

<line number> CALL "LAFONT",font number

Purpose

The LAFONT routine specifies an alternate print font. When characters are displayed using one of the Character Display routines, the main print font and the alternate print font may be used. The alternate font may be any of the thirteen available print fonts.

Examples

```
400 CALL "LMFONT",0
410 CALL "LAFONT",11
```

This example would allow display of a character string containing both normal "ASCII" characters and engineering symbols or Greek characters.

Explanation

Any non-integer font number values used as the argument in this routine are rounded to an integer from 0 to 12 inclusive. Values greater than twelve or less than zero default to Font 0. Refer to the preceding figures for descriptions of the thirteen available character fonts.

The Character Display routines discussed in following pages accept a character string as an argument. The use of certain control characters within these character strings allow switching between a main and alternate character font. Normally the font specified by the LMFONT routine is used. Control-N dictates that the following characters are to be displayed by the alternate font specified by the LAFONT routine. Control-O switches the current character font back to the main font set by the LMFONT routine.

The preceding example set the main and alternate fonts. The following line defines a character string which uses characters from the two character fonts. (Control characters are represented by N or O.)

```
430 LET A$="(cosNpOx)/(3Nc)"
```

When this character string is displayed by one of the Character Display routines, the following symbols would appear:

```
(cos πx)/(3ψ)
```

The main font is automatically reselected any time a Character Display routine is called. Even though the example above ends in the alternate character font, the next string displayed starts assuming the main character font is intended.

NOTE

The LCENTR routine is a Character Display routine which accepts a single character rather than a character string as an argument. The character font set by LAFONT cannot be called by the LCENTR routine.

Character Resolution

The Character and Symbol ROM Pack includes two routines to adjust character resolution. Character resolution becomes increasingly significant as character size is increased.

- LQUAL calls for either normal or enhanced character quality. The routine only sets one or the other; it does not provide a continuous range of quality settings.
- LSMOTH sets a continuous range of "smoothness" for characters. This routine is only activated if LQUAL is set for enhanced character resolution.

Enhanced character resolution can improve the visual quality of graphic output. However, it also takes additional time to display enhanced quality characters.

The LQUAL Routine

Syntax Form:

<line number> CAL "LQUAL",numeric expression

Descriptive Form:

<line number> CALL "LQUAL",factor

Purpose

The LQUAL routine allows setting of enhanced character resolution.

Examples

```
250 CALL "LQUAL",1
350 CALL "LQUAL",0
```

Line 250 calls for normal character resolution, as would normally be expected on the 4052 or 4054. Line 350 calls for enhanced character resolution.

Explanation

Higher character resolution means that rounded characters or portions of characters are more smoothly rounded. With normal size characters, character resolution differences may be unnoticeable. When character size is increased, differences in resolution become apparent. The price of this increased resolution is that displays take more time, whether character size is large or small.

The LQUAL routine can only have two values, on and off. The argument for the routine may be any real number. Any non-zero expression is rounded to 1, which is the default or initial LQUAL, normal quality.

The LSMOTH Routine

Syntax Form:

<line number> CAL "LSMOTH",numeric expression

Descriptive Form:

<line number> CALL "LSMOTH",factor

Purpose

The LSMOTH routine allows regulation of the degree of character smoothness.

NOTE

The LSMOTH setting is ignored when the LQUAL routine is set to one.

Examples

```
250 CALL "LQUAL",0
260 CALL "LSMOTH",1

350 CALL "LQUAL",0
360 CALL "LSMOTH",0
```

Line 250 and 350 both call for increased character resolution. Lines 260 and 360 control how much greater the resolution actually is. Lines 250 and 260 call for enhanced quality, but relatively little additional smoothness. Lines 350 and 360 call for maximum smoothness.

Explanation

The LSMOTH routine can accept argument values between 0 and 1 inclusive. One calls for normal smoothness, and zero calls for maximum smoothness. Values greater than one are rounded to one, values less than zero are rounded to zero. The degree of smoothness obtained for values between zero and one is not linear.

If enhanced LQUAL is selected, the setting of the LSMOTH routine also affects the display time. The greater the smoothness specified, the longer character display takes.

The default or initial value for the LSMOTH routine is 0.77. Remember that LSMOTH is ignored unless the LQUAL routine is set for high quality.

Cursor Position

The Character and Symbol ROM Pack includes two routines to position the cursor.

- LMOVE moves the cursor by spaces and lines. The space and line size, or character envelope size, are determined from the settings of the Character Size and Spacing routines discussed earlier.

- LHOME positions the cursor at the first print position on the page. Because the character envelope used in this ROM pack is different from the character cell as used in the standard 4052/4054, the LHOME routine places the cursor in a different position from the HOME command.

Both the LMOVE and the LHOME routines accept an optional device or I/O address. The default address is for device 32, the display screen. Only the primary address may be given, and the @ is not used.

The LMOVE Routine

Syntax Form:

<line number> CAL "LMOVE", <I/O address;> numeric expression, numeric expression

Descriptive Form:

<line number> CALL "LMOVE", <I/O address;> # spaces, # lines

Purpose

The LMOVE routine moves the cursor by units of spaces and lines. The distance moved depends on the setting of character spacing size, rather than on user data units alone.

Explanation

The LMOVE routine is designed as a convenience in positioning text. It allows the current print position to be adjusted on the same size scale as the text. In contrast, the MOVE or RMOVE statements use user data units. To RMOVE a number of lines, for example, would require multiplying the number of lines by the line size. The LMOVE routine automatically makes that conversion.

The LMOVE routine accepts real numbers as arguments. Negative values may be used for overstriking or linefeeds. Positive arguments move the cursor to the right and up. LMOVE of lines is the opposite of linefeeds.

NOTE

The LMOVE routine assumes uniform spacing. Caution is necessary when using this routine with the LETEVN routine.

The LHOME Routine

Syntax Form:

<line number> CAL "LHOME" <,I/O address>

Descriptive Form:

<line number> CALL "LHOME" <,I/O address>

Purpose

The LHOME routine moves the cursor to the upper left or first character print position.

Explanation

The LHOME routine takes the set character size into account when homing the cursor. The cursor is moved to the far upper left corner of the window, and then down by the height of an average uppercase letter. Using the LHOME routine ensures that the entire character fits onto the screen when special character sizes are called. If the character is too large to fit in the window, then the character is clipped.

Character Display

The Character and Symbol ROM Pack includes four routines which display characters using all of the parameters and routines already discussed.

- LETTER is the basic display command. It displays a character or string using all of the Size and Spacing, Rotate and Slant, Character Fonts, and Character Resolution routines.
- LCENTR uses all the features of LETTER. It only displays one character, however. That character is centered on the current cursor position.
- LVERTI uses all the features of LETTER. While LETTER displays characters side to side, LVERTI displays them top to bottom. The LVERTI routine displays a string of characters right side up, but down a vertical axis.
- LETEVN uses all the features of LETTER. It also adjusts the size of the character envelope for certain narrow characters. The space between actual characters is more uniform.

All the Character Display routines accept an optional device or I/O address. The default address is 32, the display screen.

I/O address calls for the primary GPIB address, without the '@.'

The LETTER Routine

Syntax Form:

<line number> CAL "LETTER",<I/O address;> character string

Descriptive Form:

<line number> CALL "LETTER",<I/O address;> print string

Purpose

The LETTER routine works like the normal PRINT command. Unlike the PRINT command, the LETTER routine uses the controls set by other routines included in this package.

Examples

```
430 LET A$ ="Character String"
450 CALL "LETTER",A$
```

Line 450 displays the string A$, defined in line 430, on the display screen.

```
550 CALL "LETTER",1;A$
```

Line 550 displays the string A$ on device number 1 (usually the plotter).

```
640 MOVE 10,30
650 CALL "LETTER","Number of Units"
```

Line 650 displays the characters 'Number of Units' on the display screen. The string begins at position (10,30).

Explanation

Activating all of the Character Transform routines, the LETTER routine causes any of the 96 printing ASCII characters contained in the string to be sent to the device specified. (These ASCII characters include ADE values 32 - 127, except DEL.) These characters may be displayed in any of the thirteen available type fonts (refer to the Character Fonts discussion).

The following control characters may also be included in the print string. They are not displayed, but cause some action to occur.

Character	Action
BEL (Control-G)	Rings the bell.
BS (Control-H)	Back space.
HT (Control-I)	Horizontal tab (single space).
LF (Control-J)	Line feed.
VT (Control-K)	Vertical tab (single line).
FF (Control-L)	Form feed.
SO (Control-N)	Alternate font, as selected by the LAFONT routine.
SI (Control-O)	Switches back to the standard font, as specified by the LMFONT routine.
Control-Rubout	Carriage return/line feed.

The display starts at the current cursor position. This position becomes the lower left corner of the first character in the string. After displaying the string, the cursor is positioned at the lower right of the space following the last character of the string.

The LCENTR Routine

Syntax Form:

<line number> CALL "LCENTR",<I/O address;> string expression

Descriptive Form:

<line number> CALL "LCENTR",<I/O address;> print string

Purpose

The LCENTR routine has the same characteristics as the LETTER routine. It is a PRINT command that uses all of the Character Transforms. While the LETTER routine displays a character string starting at the cursor position, the LCENTR routine displays a single character, centered on the current cursor position.

Examples

```
300 LET A$="A"
310 CALL "LSIZE",10,10
320 CALL "LQUAL",0
330 CALL "LCENTR",A$
```

Figure 2-10 shows the resulting display.

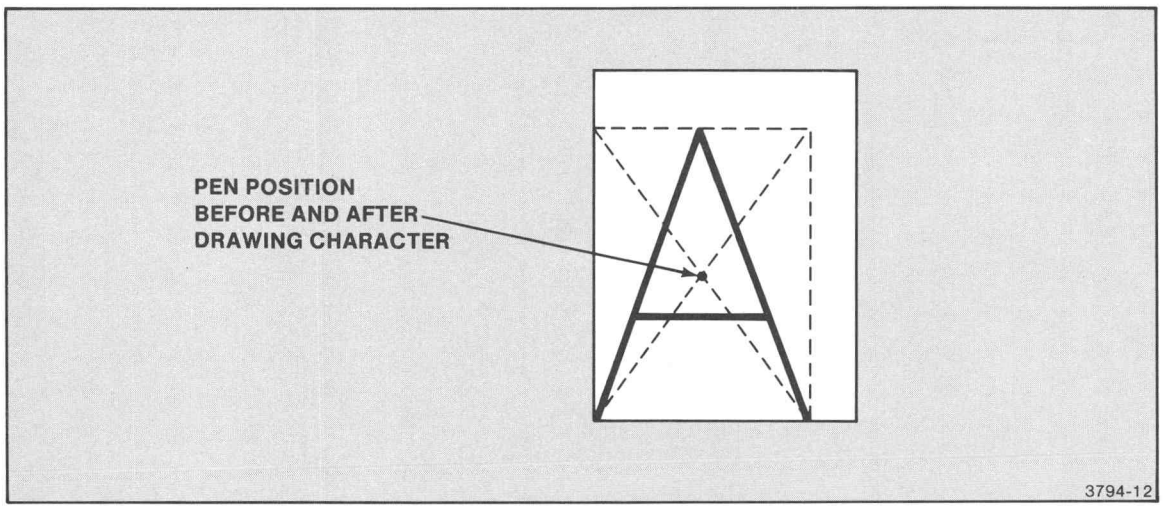


Figure 2-10. The LCENTR Routine Example.

Explanation

The LCENTR routine displays an average uppercase letter centered on the current cursor position. Lowercase letters may appear centered slightly below the current cursor position.

The cursor returns to the starting center position after the character is drawn. To print a string of characters using the LCENTR routine and not overlap them, the cursor must be moved. The LMOVE routine can be useful when interspersed between LCENTR routines.

The character string argument may include more than one character. The LCENTR routine only displays the first character of the string. Because the LCENTR routine only displays a single character, nothing appears if the string begins with a control character. Hence, characters cannot be displayed in the alternate print font with the LCENTR routine.

The LVERTI Routine

Syntax Form:

<line number> CALL "LVERTI",<I/O address;> character string

Descriptive Form:

<line number> CALL "LVERTI",<I/O address;> print string

Purpose

The LVERTI routine prints a character string top to bottom, with characters right side up. It is useful for printing labels on the vertical axis of a graph. It also uses the other Character Transform routines in the Character and Symbol ROM Pack.

Example

```
100 INIT
110 MOVE 50,80
120 CALL "LQUAL",1
140 CALL "LVERTI","The LVERTI Routine_The LVERTI Routine"
150 END
```

Figure 2-11 contains the resulting display.

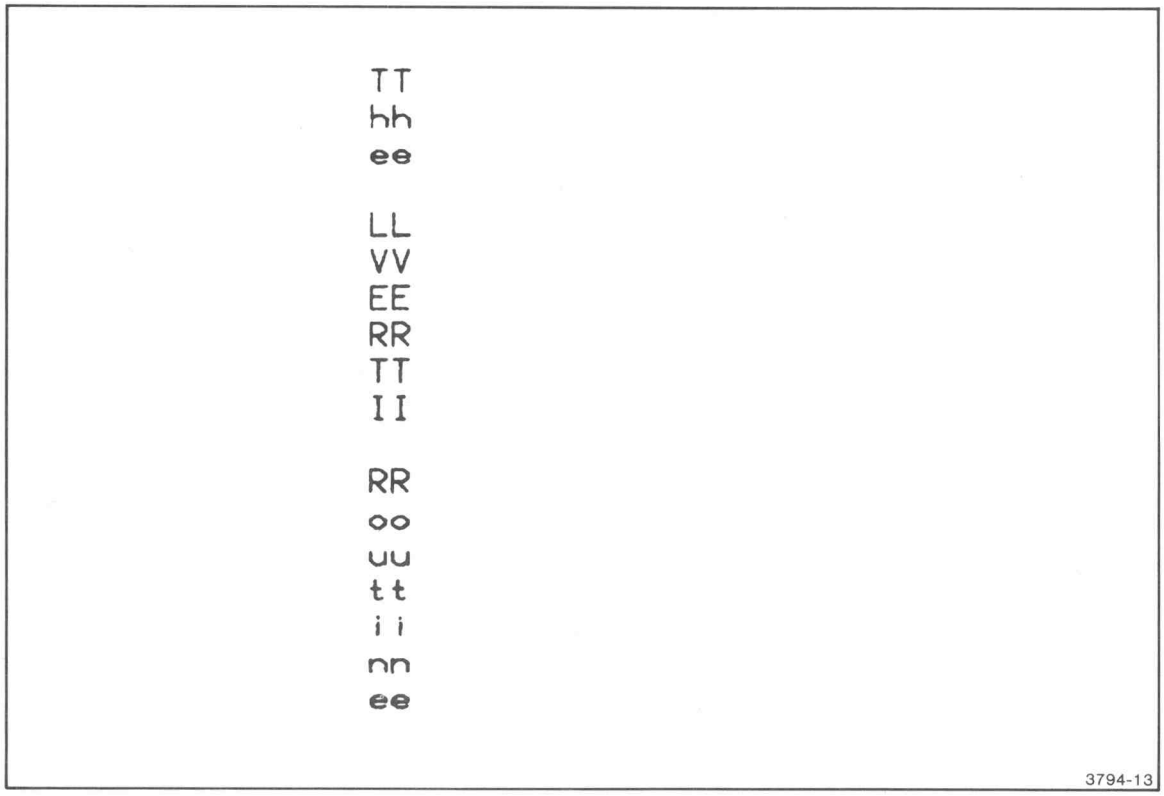


Figure 2-11. The LVERTI Routine Example.

Explanation

The LVERTI routine draws a character or string top to bottom. A line feed is a space to the right, and a carriage return takes the cursor to the top of the next column. Character size and envelope size or spacing are not affected, however. A wide character size remains wide whether the LVERTI routine or the LETTER routine are used.

The LSLANT routine slants characters displayed by the LVERTI routine. When the LROT routine is set, characters are rotated the appropriate degree from the vertical axis, rather than from the horizontal axis. The rotation point is the lower left corner of the first character.

The LETEVN Routine

Syntax Form:

<line number> CAL "LETEVN",<I/O address;> character string

Descriptive Form:

<line number> CALL "LETEVN",<I/O address;> print string

Purpose

The LETEVN routine is a Character Display routine in which the string is displayed with the spacing between characters adjusted. The character envelope for narrow characters is narrower.

Examples

```
490 MOVE 20,60
500 CALL "LSCALE",2,2
510 CALL "LETTER","Limit 1: Alt."
520 CALL "LMOVE",-13,-1
530 CALL "LETEVN","Limit 1: Alt."
540 END
```

The example displayed in Figure 2-12 shows one string with normal spacing and another string with the spacing between characters adjusted.

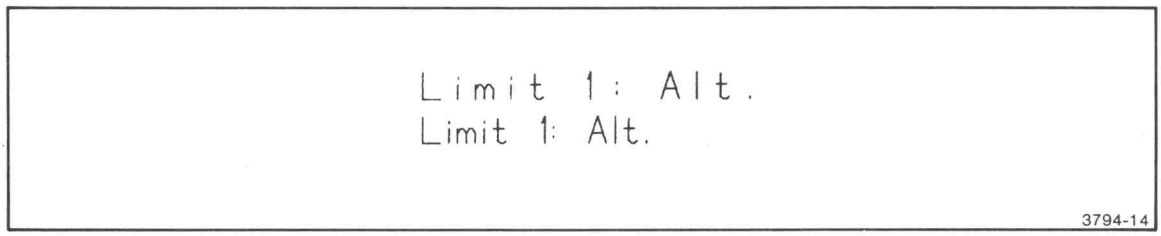


Figure 2-12. The LETEVN Routine Example.

Explanation

The LETEVN routine is identical to the LETTER routine except that narrower spaces are allotted for certain characters. These characters are:

```
FONTS 0,3,6,7,and 8: I 1 : ; . , ! ' ` | i l t
FONTS 1,2,5,and 9: I 1 : ; . , ! ' ` i l t
FONT 4:I 1 : i ; . , ! ' ` | i l t
FONT 10:1 : ; . , ! ' ` ∂ αβδεφγιξκλνοστςχζ
FONT 11: ° : ‖ . £ · ∂ αβδεφγιξκλνοστςχζ
FONT 12: I 1 : ° ; . , ! ' ` i l t
```

The LRESET Routine

Syntax Form:

<line number> CALL "LRESET"

Purpose

The LRESET routine sets all parameters included in this package to their initial state.

Explanation

The LRESET routine puts all the values set by the routines in the Character and Symbol ROM Pack to their initial values. It does not affect any other values. These values may also be initialized with the INIT statement.

The LRESET routine effects the following settings.

Table 2-1

INITIAL ROUTINE SETTINGS

Command	Initial Setting
LDIM	1.71, 2.29 user data units
LAFONT	0
LMFONT	0
LQUAL	1
LRATIO	1.5, 1.636
LROT	0
LSCALE	1,1
LSIZE	1.14, 1.79 user data units
LSLANT	0
LSMOTH	0.77
SMOOTH	0.5

ARCS AND CIRCLES

The Character and Symbol ROM Pack includes three routines for drawing arcs and circles, and one for setting the resolution or smoothness of those arcs and circles.

- ARC1 draws an arc or circle:
  Around a given point;
  Starting at the current cursor position;
  Extending around for a given angle.

- ARC2 draws an arc or circle:
  Around a given point;
  With a given radius;
  From one angle to another.

- CIRCLE draws a circle:
  With a given radius;
  Around the current cursor position.

- SMOOTH sets the degree of smoothness of any arcs or circles.

Angles called in these routines depend on the environmental settings. Degrees, radians, or grads may be set.

Arcs and circles are affected by the window setting. They are calculated in user data units. If the default window ratio (130/100) is maintained, normal arcs and circles are displayed. If a distorted window is specified, the arcs and circles are also distorted. Figure 2-13 shows how a distorted window causes an ellipse to be drawn.

```
100 WINDOW 0,10,0,100
110 MOVE 5,70
120 CALL "CIRCLE",5
130 END
```

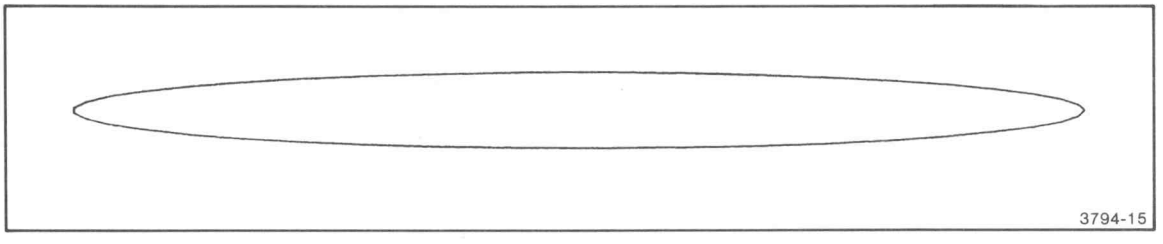


Figure 2-13. How To Draw an Ellipse.

Device or I/O addresses may be given within ARC1, ARC2, and CIRCLE routines. The normal default device is 32, the display screen.

I/O address calls for the primary GPIB address, with no '@.'

The ARC1 Routine

Syntax Form:

<line number> CAL "ARC1",<I/O address;> numeric expression, numeric expression, numeric expression

Descriptive Form:

<line number> CALL "ARC1",<I/O address;> center x, center y, angle

Purpose

The ARC1 routine draws an arc. The center of the arc is specified in the argument. The starting position of the arc is the current cursor position. The angle argument describes the length of the arc.

Examples

```
700 SET DEGREES
710 MOVE 10,10
720 CALL "ARC1",20,30,120
```

Figure 2-14 describes how the ARC1 routine draws an arc.

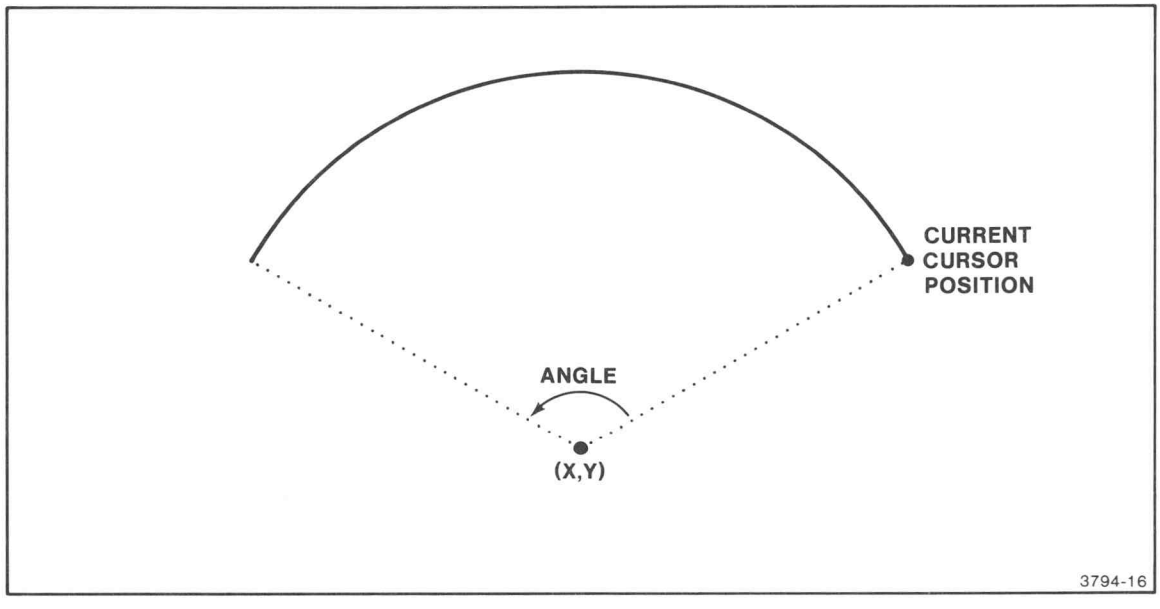


Figure 2-14. The ARC1 Routine.

Explanation

The ARC1 routine works in the following manner:

1. A vector is calculated (not drawn) between the current cursor position and the point specified by the x and y arguments.

2. A vector of equal length is calculated at the angle specified from the first vector. Positive angles are measured counterclockwise from the first vector, while negative angles are figured clockwise.

3. A best fit arc is drawn between the ends of each vector. The smoothness or fit depends on the setting of the SMOOTH routine.

The ARC2 Routine

Syntax Form:

<line number> CAL "ARC2",<I/O address;> numeric expression, numeric expression, numeric expression, <numeric expression, numeric expression>

Descriptive Form:

<line number> CALL "ARC2",<I/O address;> center x, center y, radius, <starting angle, ending angle>

Purpose

The ARC2 routine displays an arc of specified radius centered around a specified point. The ends of the arc are described by angles from this center point. If angles are not given, the ARC2 routine draws a circle.

Examples

```
700 SET DEGREES
800 CALL "ARC2",50,60,10,-30,90
```

Figure 2-15 describes how the ARC2 routine draws an arc.

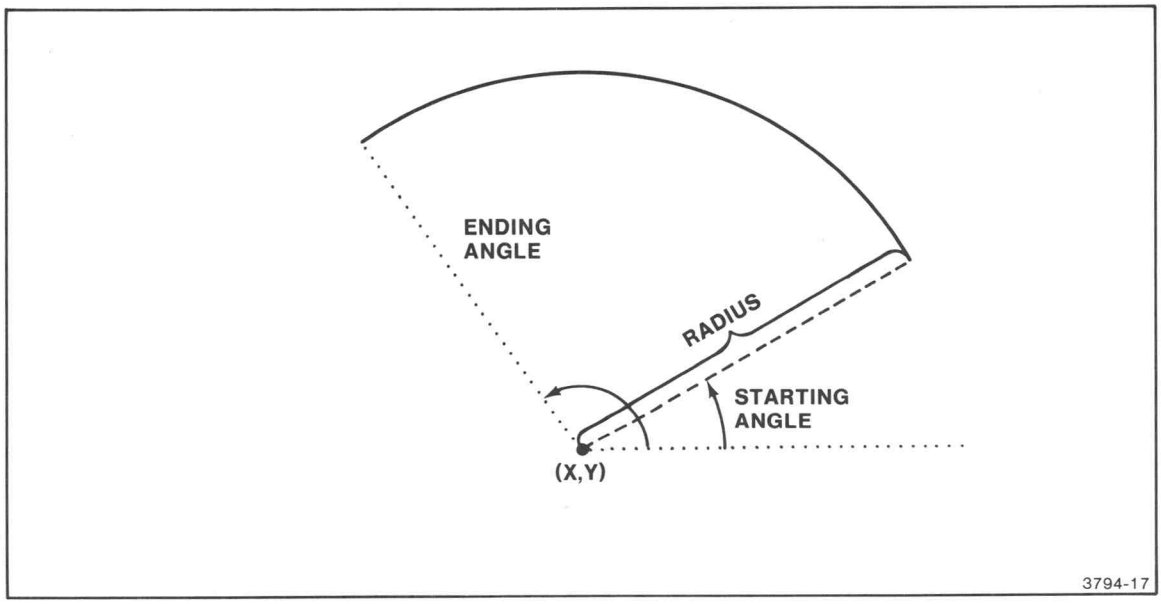


Figure 2-15. The ARC2 Routine.

Explanation

The starting and ending angles are measured from the horizontal vector extending to the right of the center point. Positive angles extend counterclockwise, and negative angles are measured clockwise from this point. The arc is drawn counterclockwise from the starting point to the end point.

The CIRCLE Routine

Syntax Form:

<line number> CAL "CIRCLE",<I/O address;> numeric expression

Descriptive Form:

<line number> CALL "CIRCLE,<I/O address;> radius

Purpose

The CIRCLE routine draws a circle of specified radius. The circle is centered around the current cursor position.

Examples

```
800 MOVE 50,80
810 FOR X1=1 TO 10
820 CALL "CIRCLE",X1
830 NEXT X1
840 END
```

Figure 2-16 shows the resulting display.

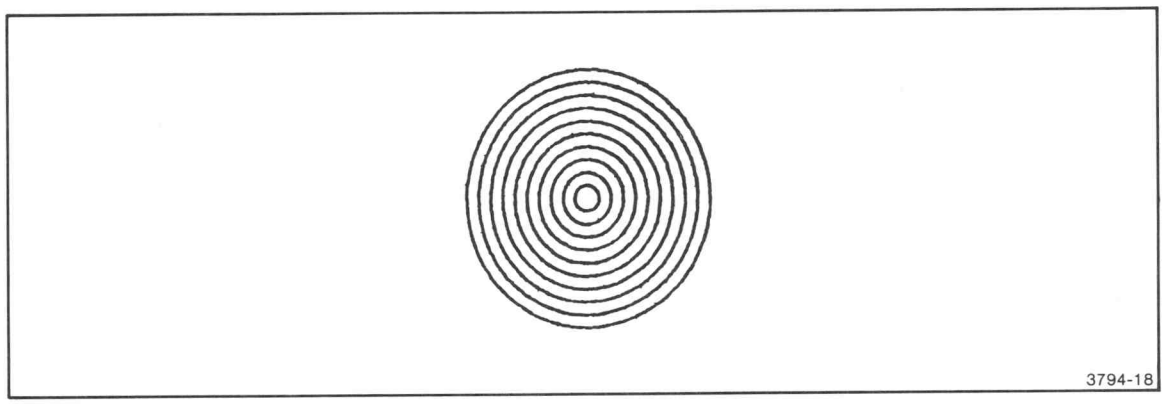


Figure 2-16. The CIRCLE Routine Example.

The SMOOTH Routine

Syntax Form:

<line number> CAL "SMOOTH", numeric expression

Descriptive Form:

<line number> CALL "SMOOTH", factor

Purpose

The SMOOTH routine establishes a resolution level for arcs and circles. Arcs and circles are drawn as a series of vectors or straight lines. This routine determines the number of vectors used; the more vectors, the smoother the curve appears.

Examples

```
1230 CALL "SMOOTH",1
1240 CALL "CIRCLE",10
```

This first example displays an octagon with a maximum diameter of 20 units.

```
1330 CALL "SMOOTH",0
1340 CALL "CIRCLE",10
```

This example draws a smooth circle on the display screen. The curvature should appear uniform.

Explanation

The SMOOTH routine only affects the ARC1, ARC2, and CIRCLE routines.

The SMOOTH routine uses values between zero and one. Arguments less than zero are set to zero. Arguments greater than one are set to one. Arguments between zero and one give varying degrees of smoothness, with one being least smooth and zero being most smooth. The variation in smoothness between one and zero is not linear, however. The default or initial value of SMOOTH is 0.5. This setting provides a relatively high level of smoothing.

Increased smoothness requires longer display times.

The LRESET routine resets the SMOOTH function to the initial value (0.5).

CHARACTER AND SYMBOL GENERATION

The STROKE Routine

Syntax Form:

<line number> CALL "STROKE",<I/O address;>string variable,array variable

Descriptive Form:

<line number> CALL "STROKE",<I/O address;> command string,argument array

Purpose

The STROKE routine draws special figures, symbols, characters, designs, or logos. The symbol drawn by the STROKE routine is subject to all of the Character Transforms in the Character and Symbol ROM Pack.

Examples

```
1700 SET DEGREES
1710 A$="MCCMDDDMDD1DDDDDDMDD1DDDDDE"
1720 DIM A(50)
1730 DATA 5,5,5,1,4,9.9,4,17,6,17,6,9.9,4,12
1740 DATA 1,12,1,9.5,0.5,9.5,-180,0,20,1,20,1,12,1,13,4,13,6,12
1750 DATA 9,12,9,9.5,9.5,9.5,180,10,20,9,20,9,12,9,13,6,13
1760 READ A
1770 MOVE 50,20
1780 CALL "LSIZE",6,11
1790 CALL "STROKE",A$,A(1)
1800 CALL "LROT",45
1810 RMOVE 10,10
1820 CALL "STROKE",A$,A(1)
1830 END
```

Figure 2-17 contains the resulting display.

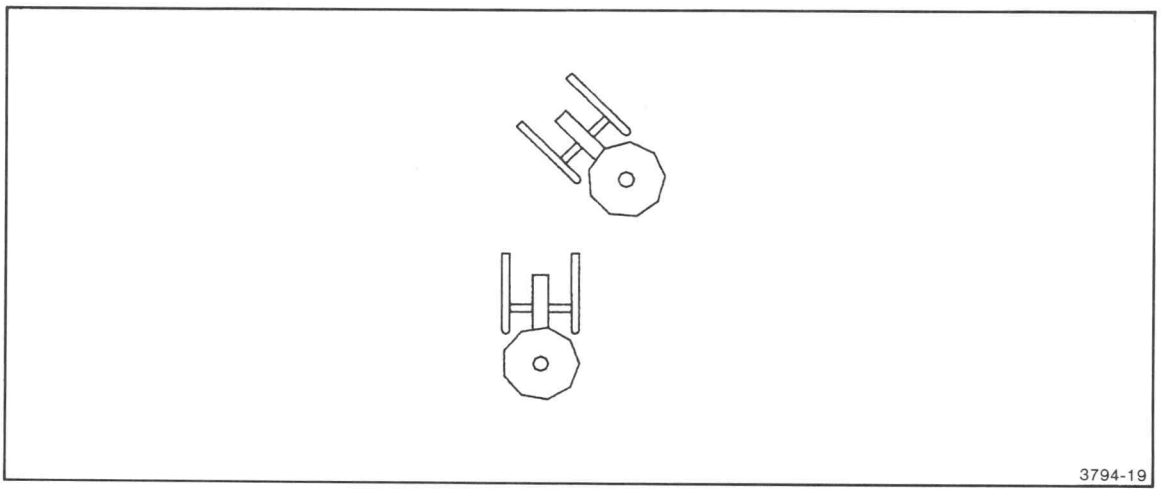


Figure 2-17. The STROKE Routine Example.

Explanation

The STROKE routine requires two arguments. The first is a string variable which gives a command string. The second is an array which gives the arguments required by the command string.

The STROKE routine sequentially executes the commands listed within the command string. Each command in turn sequentially pulls any arguments that the command requires from the argument array.

Command String

The command string is a list of one character commands. These commands represent each step taken to draw the character. The valid commands are:

M — Move to the argument point. Two data arguments are required to define the argument point.

D — Draw a line from the current point to the argument point. Two data items are required to define the argument point.

1 — Draw an arc as would be specified by the ARC1 routine. As in the ARC1 routine, three arguments are required. Two set the center point for the arc, and the third is an angle describing the arc length. The prior cursor position is the starting point of the arc.

2 — Draw an arc as would be specified by the ARC2 routine. Because of the string aspect of the STROKE routine, all five arguments must be present. The first two data items define the center point of the arc, the third describes the radius, and the fourth and fifth specify the starting and ending angles.

C — Draw a circle centered on the current cursor position, with a radius of the single data argument, as is done with the CIRCLE routine.

E — End the character definition. No data item is used.

Data Argument Arrays

As shown in the command string explanation, each command requires specific arguments. The STROKE routine checks the array length against the command string. The array must contain the exact number of elements required by the command string.

The arguments are all relative to the last point moved or drawn to before the STROKE routine is called. That starting point becomes (0,0) for the STROKE routine arguments.

Note that the STROKE routine calls an array argument with a subscript of 1.

Designing Symbols

Analyzing the preceding example can further explain the STROKE routine. The command string, A$, is given in line 1710. Data for the argument array is given in lines 1730, 1740, and 1750. Combining the two, the STROKE command in effect executes the following program:

```
MOVE 5,5
CALL "CIRCLE",5
CALL "CIRCLE",1
MOVE 4,9.9
DRAW 4,17
DRAW 6,17
DRAW 6,9.9
MOVE 4,12
DRAW 1,12
DRAW 1,9.5
CALL "ARC1",0.5,9.5,-180
DRAW 0,20
DRAW 1,20
DRAW 1,12
DRAW 1,13
DRAW 4,13
MOVE 6,12
DRAW 9,12
DRAW 9,9.5
CALL "ARC1",9.5,9.5,180
DRAW 10,20
DRAW 9,20
DRAW 9,12
DRAW 9,13
DRAW 6,13
END
```

The STROKE routine is subject to all of the Character Transform routines described earlier in this section. The character or symbol drawn by the STROKE routine may be slanted or rotated and redimensioned by the character routines in this ROM pack.

For example, note line 1800 above, the CALL "LSIZE",6,11. This line is required to convert the character envelope to user data units. If this width-to-height aspect is changed from 6/11 (the normal character envelope proportions), arcs and circles become elliptical.

Smoothing of arcs and circles drawn by the STROKE routine is controlled by the LQUAL and LSMOTH routines, and not by the SMOOTH routine.

ENHANCEMENTS TO GRAPHICS

The Character and Symbol ROM Pack includes four routines which provide additional graphics capabilities.

- TYPEKY enhances the POINTER command. It inserts a carriage return in the typeahead buffer. The POINTER command uses this character and does not stop for input.

- CURSOR enables the crosshairs and returns the current cursor position. When used with 4054 Option 30, CURSOR also enables the drag bit.

- ENDRAG allows refresh objects to be dragged with the thumbwheels of a 4054 while a program is executing. The ENDRAG routine has no effect when called from a 4052 or 4054 without Option 30, Dynamic Graphics.

- CLDRAG turns off the drag bit and clears ENDRAG. It has no effect when used with a 4052 or 4054 without Option 30.

The TYPEKY Routine

Syntax Form:

<line number> CAL "TYPEKY"

Purpose

The TYPEKY routine makes the 4052/4054 POINTER routine return the current position of the cursor without a carriage return.

Explanation

The TYPEKY routine allows the POINTER command to be used within a program. The program does not wait for the operator to enter a carriage return; the TYPEKY routine does it automatically. A carriage return is given as POINTER input; the POINTER command is answered by the program instead of the user.

The POINTER removes the carriage return from the typeahead buffer. The typeahead buffer is also affected by other operations. To use this feature, the POINTER command should be immediately preceded by CALL "TYPEKY".

The CURSOR Routine

Syntax Form:

<line number> CAL "CURSOR", variable, variable

Descriptive Form:

<line number> CALL "CURSOR", x-coordinate destination, y-coordinate destination

Purpose

The CURSOR routine enables the crosshairs and returns the current position of the cursor. It is equivalent to POINTER <CR>. It tracks the thumbwheel setting of the 4054 and the joystick position on the 4052.

Example (For 4054 Option 30 Only)

```
130 MOVE 50,50
140 ROPEN 1
160 RDRAW 5,-5
170 RDRAW -10,0
180 RDRAW 5,5
190 RCLOSE
200 CURSOR 1
230 CALL "CURSOR",X1,Y1
240 CALL "WAIT",0.3
250 X=X1
260 Y=Y1
270 MOVE 30,80
280 ROPEN 9
290 PRINT USING "3D.2D,3X,3D.2D":X,Y;
300 RCLOSE
310 GO TO 200
```

This short program draws a movable pointer (lines 110 — 160). The CURSOR routine (line 170) captures the coordinates of the current point. These coordinates are continually reported (lines 220 — 240) as the cursor is moved.

Explanation

The CALL "CURSOR",X,Y statement is also equivalent to:

```
CALL "TYPEKY"
POINTER X,Y,Z$
```

The CURSOR routine also enables the drag bit. Refer to the next routine, the ENDRAG routine.

The ENDRAG Routine

Syntax Form:

<line number> CAL "ENDRAG"

Purpose

NOTE

The ENDRAG routine only works with 4054 Option 30.

The ENDRAG (enable drag) routine allows refresh objects to be dragged with the thumbwheels while a program is executing. The ENDRAG routine turns on the current cursor object. By turning on multiple objects, multiple rubber-banding is possible.

Examples

```
1001 REM MULTIPLE SEGMENT RUBBERBANDING
1100 INIT
1110 DIM X(8),Y(8)
1120 DATA 0,65,130,130,130,65,0,0
1130 DATA 0,0,0,50,100,100,100,50
1140 READ X
1150 READ Y
1160 FOR I=1 TO 8
1170 MOVE 65,50
1180 ROPEN I
1190 VISIBILITY I,0
1200 DRAW X(I),Y(I)
1210 RCLOSE
1220 NEXT I
1230 PAGE
1240 FOR I=1 TO 7
1250 CURSOR I
1260 CALL "ENDRAG"
1270 VISIBILITY I,1
1280 NEXT I
1290 VISIBILITY 8,1
1300 CURSOR 8
1310 POINTER X0,Y0,I$
1320 IF LEN(I$) THEN 1340
1330 END
1340 FOR I=1 TO 8
1350 FIX I
1360 NEXT I
1370 GO TO 1310
```

This program demonstrates how multiple segments or objects may be dragged.

Lines 1110 thru 1150 set eight x- and y-coordinate arrays of points, four at the display corners and four at the display side midpoints.

Lines 1160 thru 1220 create a set of objects. Each object is a vector from one of the points set earlier to the center of the display. The center point is arbitrarily chosen; any point may be used. All the objects which are dragged must converge on one drag point.

Lines 1240 thru 1300 tie all eight objects together and set visibility.

Line 1310 enables the thumbwheels to control the CURSOR position for objects 1 thru 8.

Lines 1320 thru 1370 "fix" the eight vectors in position when a keyboard character is entered. A carriage return ends the program.

The CLDRAG Routine

Syntax Form:

<line number> CAL "CLDRAG"

Purpose

NOTE

The CLDRAG routine only works with 4054 Option 30.

The CLDRAG routine cancels or clears the ENDRAG setting for the current cursor function.

Section 3
REPLACEABLE PARTS

PARTS ORDERING INFORMATION

Replacement parts are available from or through your local Tektronix, Inc. Field Office or representative.

Changes to Tektronix instruments are sometimes made to accommodate improved components as they become available, and to give you the benefit of the latest circuit improvements developed in our engineering department. It is therefore important, when ordering parts, to include the following information in your order: Part number, instrument type or number, serial number, and modification number if applicable.

If a part you have ordered has been replaced with a new or improved part, your local Tektronix, Inc. Field Office or representative will contact you concerning any change in part number.

Change information, if any, is located at the rear of this manual.

SPECIAL NOTES AND SYMBOLS

X000	Part first added at this serial number
00X	Part removed after this serial number

FIGURE AND INDEX NUMBERS

Items in this section are referenced by figure and index numbers to the illustrations.

INDENTATION SYSTEM

This mechanical parts list is indented to indicate item relationships. Following is an example of the indentation system used in the description column.

1 2 3 4 5 Name & Description

Assembly and/or Component
Attaching parts for Assembly and/or Component
- - - * - - -
Detail Part of Assembly and/or Component
Attaching parts for Detail Part
- - - * - - -
Parts of Detail Part
Attaching parts for Parts of Detail Part
- - - * - - -

Attaching Parts always appear in the same indentation as the item it mounts, while the detail parts are indented to the right. Indented items are part of, and included with, the next higher indentation. The separation symbol - - - * - - - indicates the end of attaching parts.

Attaching parts must be purchased separately, unless otherwise specified.

ITEM NAME

In the Parts List, an Item Name is separated from the description by a colon (:). Because of space limitations, an Item Name may sometimes appear as incomplete. For further Item Name identification, the U.S. Federal Cataloging Handbook H6-1 can be utilized where possible.

ABBREVIATIONS

"	INCH
#	NUMBER SIZE
ACTR	ACTUATOR
ADPTR	ADAPTER
ALIGN	ALIGNMENT
AL	ALUMINUM
ASSEM	ASSEMBLED
ASSY	ASSEMBLY
ATTEN	ATTENUATOR
AWG	AMERICAN WIRE GAGE
BD	BOARD
BRKT	BRACKET
BRS	BRASS
BRZ	BRONZE
BSHG	BUSHING
CAB	CABINET
CAP	CAPACITOR
CER	CERAMIC
CHAS	CHASSIS
CKT	CIRCUIT
COMP	COMPOSITION
CONN	CONNECTOR
COV	COVER
CPLG	COUPLING
CRT	CATHODE RAY TUBE
DEG	DEGREE
DWR	DRAWER
ELCTRN	ELECTRON
ELEC	ELECTRICAL
ELCTLT	ELECTROLYTIC
ELEM	ELEMENT
EPL	ELECTRICAL PARTS LIST
EQPT	EQUIPMENT
EXT	EXTERNAL
FIL	FILLISTER HEAD
FLEX	FLEXIBLE
FLH	FLAT HEAD
FLTR	FILTER
FR	FRAME or FRONT
FSTNR	FASTENER
FT	FOOT
FXD	FIXED
GSKT	GASKET
HDL	HANDLE
HEX	HEXAGON
HEX HD	HEXAGONAL HEAD
HEX SOC	HEXAGONAL SOCKET
HLCPS	HELICAL COMPRESSION
HLEXT	HELICAL EXTENSION
HV	HIGH VOLTAGE
IC	INTEGRATED CIRCUIT
ID	INSIDE DIAMETER
IDENT	IDENTIFICATION
IMPLR	IMPELLER
IN	INCH
INCAND	INCANDESCENT
INSUL	INSULATOR
INTL	INTERNAL
LPHLDR	LAMPHOLDER
MACH	MACHINE
MECH	MECHANICAL
MTG	MOUNTING
NIP	NIPPLE
NON WIRE	NOT WIRE WOUND
OBD	ORDER BY DESCRIPTION
OD	OUTSIDE DIAMETER
OVH	OVAL HEAD
PH BRZ	PHOSPHOR BRONZE
PL	PLAIN or PLATE
PLSTC	PLASTIC
PN	PART NUMBER
PNH	PAN HEAD
PWR	POWER
RCPT	RECEPTACLE
RES	RESISTOR
RGD	RIGID
RLF	RELIEF
RTNR	RETAINER
SCH	SOCKET HEAD
SCOPE	OSCILLOSCOPE
SCR	SCREW
SE	SINGLE END
SECT	SECTION
SEMICOND	SEMICONDUCTOR
SHLD	SHIELD
SHLDR	SHOULDERED
SKT	SOCKET
SL	SLIDE
SLFLKG	SELF-LOCKING
SLVG	SLEEVING
SPR	SPRING
SQ	SQUARE
SST	STAINLESS STEEL
STL	STEEL
SW	SWITCH
T	TUBE
TERM	TERMINAL
THD	THREAD
THK	THICK
TNSN	TENSION
TPG	TAPPING
TRH	TRUSS HEAD
V	VOLTAGE
VAR	VARIABLE
W/	WITH
WSHR	WASHER
XFMR	TRANSFORMER
XSTR	TRANSISTOR

CROSS INDEX—MFR. CODE NUMBER TO MANUFACTURER

Mfr. Code	Manufacturer	Address	City, State, Zip
09922	BURNDY CORPORATION	RICHARDS AVENUE	NORWALK, CT 06852
01121	ALLEN-BRADLEY COMPANY	1201 2ND STREET SOUTH	MILWAUKEE, WI 53204
04222	AVX CERAMICS, DIVISION OF AVX CORP.	P O BOX 867, 19TH AVE. SOUTH	MYRTLE BEACH, SC 29577
55680	NICHICON/AMERICA/CORP.	6435 N PROESEL AVENUE	CHICAGO, IL 60645
80009	TEKTRONIX, INC.	P O BOX 500	BEAVERTON, OR 97077
83385	CENTRAL SCREW CO.	2530 CRESCENT DR.	BROADVIEW, IL 60153
91260	CONNOR SPRING AND MFG. CO.	1729 JUNCTION AVE.	SAN JOSE, CA 95112

Component No.	Tektronix Part No.	Serial/Model No. Eff	Dscont	Name & Description	Mfr Code	Mfr Part Number
A1	670-7289-00			CKT BOARD ASSY:CHAR SYMBOL ROM PACK	80009	670-7289-00
A1C1	283-0422-00			CAP.,FXD,CER DI:0.047UF,+80-20%,50V	04222	DG015E473Z
A1C13	283-0422-00			CAP.,FXD,CER DI:0.047UF,+80-20%,50V	04222	DG015E473Z
A1C15	290-0804-00			CAP.,FXD,ELCTLT:10UF,+50-10%,25V	55680	25ULA10V-T
A1C111	283-0422-00			CAP.,FXD,CER DI:0.047UF,+80-20%,50V	04222	DG015E473Z
A1R111	315-0202-00			RES.,FXD,CMPSN:2K OHM,5%,0.25W	01121	CB2025
A1U1	160-1313-00			MICROCIRCUIT,DI:4096 X 8 EPROM,PRGM	80009	160-1313-00
A1U11	160-1314-00			MICROCIRCUIT,DI:4096 X 8 EPROM,PRGM	80009	160-1314-00
A1U13	160-1315-00			MICROCIRCUIT,DI:4096 X 8 EPROM,PRGM	80009	160-1315-00
A1U15	160-1316-00			MICROCIRCUIT,DI:4096 X 8 EPROM,PRGM	80009	160-1316-00
A1U111	156-0469-02			MICROCIRCUIT,DI:3/8 LINE DCDR,BURN-IN	01295	SN74LS138

Fig. & Index No.	Tektronix Part No.	Serial/Model No. Eff	Dscont	Qty	1 2 3 4 5	Name & Description	Mfr Code	Mfr Part Number
1-1	380-0384-01			1		HSG HALF,ROM PK:LID,ABS	80009	380-0384-01
						(ATTACHING PARTS)		
-2	211-0102-00			4		SCREW,MACHINE:4-40 X 0.500",FLH,STL	83385	OBD
						- - - * - - -		
-3	367-0189-00			1		HANDLE,BOW:2.62 L,SST	91260	OBD
-4	380-0343-01			1		HSG HALF,PTR:INNER,ABS	80009	380-0343-01
-5	----- -----			1		CKT BOARD ASSY:CHAR SYMBOL ROM PACK(SEE A1 REPL)		
-6	136-0751-00			4		. SKT,PL-IN ELEK:MICROCKT,24 PIN	09922	D1LB24P-108
-7	----- -----			1		MKR SET,IDENT:MKD 4052R11 CHAR AND SYMBOL		
						STANDARD ACCESSORIES		
	070-3794-00			1		MANUAL,TECH:INSTRUCTION	80009	070-3794-00

Appendix A

VERIFICATION PROGRAM

INTRODUCTION

The following program exercises each of the routines in the 4052R11 Character and Symbol ROM Pack. This program, written in 4050 Series BASIC, also illustrates the use and syntax of the routines provided by the ROM pack. If you encounter errors while using the ROM pack, this program can be run to verify that the ROM pack is functioning properly.

This program does not use any of the Graphics Enhancement routines. Use the example programs given in Section 2 to verify that these routines function correctly.

RUNNING THE SAMPLE PROGRAM

To run the program, install the ROM pack as discussed in Section 1. Then enter the program as shown. Refer to the 4050 Series Graphic System Reference Manual if you need more information.

```
100 REM 4052R11 Verification Program
110 INIT
120 PAGE
130 SET DEGREES
140 A$="M12MCE"
150 DIM A(13)
160 DATA 10,0,20,10,-90,10,10,10,90,270,10,10,10
170 READ A
180 CALL "LSIZE",6,11
190 CALL "LQUAL",0
200 CALL "LSMOTH",0.5
210 CALL "ARC2",10,85,10
220 CALL "LMOVE",0.6,-0.55
230 CALL "STROKE",A$,A(1)
240 CALL "LMOVE",3,0
250 CALL "CIRCLE",10
260 RMOVE 0,10
270 RDRAW 0,-20
280 CALL "LMOVE",4,1.1
290 CALL "LROT",180
300 CALL "STROKE",A$,A(1)
310 CALL "LROT",0
320 CALL "LMOVE",1.8,0
330 CALL "ARC1",115,85,360
340 CALL "LSCALE",0.5,0.5
350 CALL "LHOME"
360 CALL "LMOVE",1.5,-3
370 CALL "LSLANT",10
380 CALL "LETEUN","4052R11 Verification Program"
390 MOVE 62,55
400 CALL "CIRCLE",10
410 CALL "SMOOTH",1
420 CALL "CIRCLE",10
430 CALL "LRESET"
```

```
440 CALL "LSIZE",12,16
450 CALL "LQUAL",0
460 CALL "LSMOTH",0.4
470 CALL "LCENTR","R"
480 MOVE 5,40
490 F$="   __[1234567890:-]{!"""@$%&'()0*=}_"
500 O$="`qwertyuiop@~QWERTYUIOP`_"
510 N$="asdfghjkl;\ASDFGHJKL+|__"
520 T$="zxcvbnm,./ZXCVBNM<>?_"
530 CALL "LRESET"
540 CALL "LDIM",1.3,2
550 CALL "LRATIO",1.3,1.4
560 CALL "LVERTI","Input a font number____"
570 INPUT F
580 CALL "LMFONT",F
590 CALL "LSIZE",1.3,2
600 CALL "LRATIO",1.3,2
610 CALL "LETTER",F$
620 CALL "LETTER",O$
630 CALL "LETTER",N$
640 CALL "LETTER",T$
650 MOVE 65,40
660 CALL "LRESET"
670 CALL "LDIM",1.3,2
680 CALL "LRATIO",1.3,1.4
690 CALL "LVERTI","Input Alternate font___"
700 INPUT F
710 CALL "LAFONT",F
720 F$="_   __[1234567890:-]{!"""@$%&'()0*=}_"
730 O$="_`qwertyuiop@~QWERTYUIOP`_"
740 N$="_asdfghjkl;\ASDFGHJKL+|__"
750 T$="_zxcvbnm,./ZXCVBNM<>?_"
760 CALL "LSIZE",1.3,2
770 CALL "LRATIO",1.3,2
780 CALL "LETTER",F$
790 CALL "LETTER",O$
800 CALL "LETTER",N$
810 CALL "LETTER",T$
820 END
```

NOTE

Lines 490, 500, 510, 520, 560, 690, 720, 730, 740, and 750 all contain Control-Rubout characters to provide carriage returns. This character appears as an underline, but must be entered as Control-Rubout.

After the program is entered into the 4052 or 4054 Graphic System, type RUN. If you encounter errors, refer to Appendix C. Find the error number and check the probable causes.

As the program runs, the following displays are printed on the screen. The program requests input of a font number and an alternate font number. This refers to the thirteen character fonts included in the ROM pack. When prompted with the blinking "?", enter a font number, zero through twelve, and press the carriage return. Do the same at the second prompt.

The display that appears on the screen should correspond to the one shown in Figure A-1. In this example, Font 10 and Font 12 have been selected. Different font selections should display the characters illustrated in Figures 2-6 through 2-9.

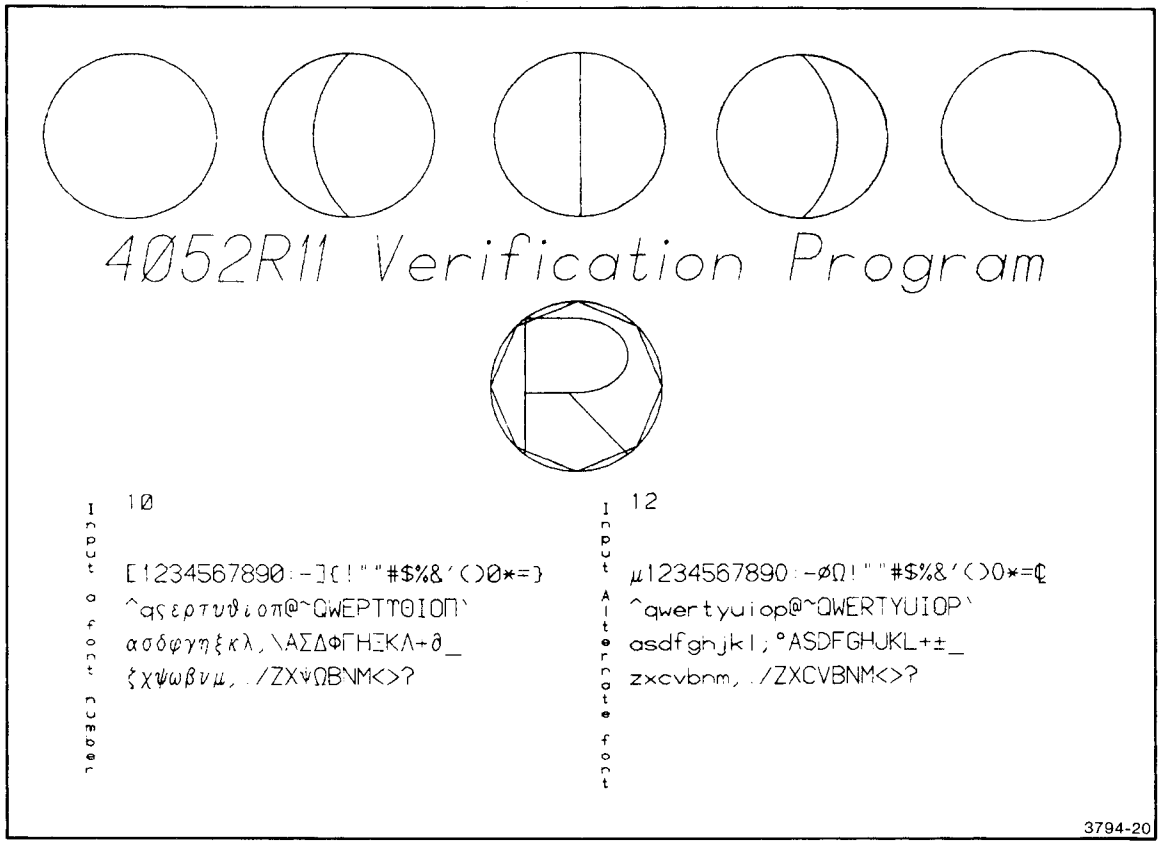


Figure A-1. Output from the Verification Program.

UNDERSTANDING THE PROGRAM

Line #	Description
110 to 130	Set initial parameters.
140 to 170	Define the command string and data array arguments for the STROKE routine. The command string tells the STROKE routine to move from the initial position to the start of an arc, to draw ARC1, to draw ARC2, to move to the center of the symbol, and to draw a circle. The circle overlaps the arc draw by the ARC2 portion. While the ARC2 portion is not essential for the symbol, it is included to test that aspect of the STROKE routine.
180 to 200	Adjust size and smoothing for the STROKE routine. Without the LSIZE setting, the arcs and circles in the STROKE routine would be oval.
210	Use the ARC2 routine to draw a circle centered at (10,85).
220 to 230	Move the cursor and draw the STROKE symbol. Line 210 leaves the cursor on the right-most edge of the circle. Line 220 moves the cursor over and down to the origin, or lower-left corner, of the character cell used for the STROKE command.
240 to 270	Draw a circle with a vertical line through it. Line 230 leaves the cursor in the center of the symbol (the last portion drawn is the circle). Line 240 moves the cursor horizontally to the center of the circle drawn in line 250. After line 270, the cursor is at the bottom of the circle.
280 to 300	Move the cursor and draw the STROKE symbol, rotated 180°. Line 280 moves the cursor to the upper right of the character cell used for the symbol, rather than to the normal origin. Line 290 dictates that the symbol be rotated 180° around the cursor position, transforming the symbol into proper position.
310 to 330	Position the cursor and draw a circle with the ARC1 routine. The rotation must first be returned to zero. (Otherwise line 320 would move the cursor to the left.)

340 to 380	Print the string "4052R11 Verification Program" below the line of circles. The proportioning for this string is the same as with the previous characters (the LSIZE setting of 6,11, standard character envelope proportions). LSCALE, line 340, preserves this ratio but diminishes the size. Lines 350 and 360 position the cursor three lines down (the circles take up two of these resized lines) and 1.5 spaces in. Line 370 slants the characters, and line 380 displays the string, with the space between characters adjusted.
390 to 420	Demonstrate the SMOOTH function. Line 400 calls a circle with default smoothing (0.5). Line 420 calls a circle with no smoothing.
430 to 470	Demonstrate the LCENTR routine. Line 430 cancels all transform settings, particularly the LSLANT. The following lines set size and smoothing, and line 470 centers a character around the current cursor position, the center of the circle.
490 to 520	Define character strings for subsequent font displays. Note that the two underlines in the beginning of the string in line 490, and at the end of lines 490, 500, 510, and 520 are Control-Rubout characters; they signal linefeeds. Also note the multiple quotes required in line 490.
530 to 560	Reset character size and demonstrate the LVERTI routine.
570 to 640	Input a font number, reset the character size, and display the font. Any number may be input, though any number except one through twelve will default to font zero. The LETTER routine cannot reference multiple strings; a separate call must be made for each string. Each string begins at the current cursor position. A carriage return, if needed, must be embedded in the string itself. Hence the Control-Rubout characters in lines 490 thru 520.
650 to 690	Reset the character dimensions and use the LVERTI routine to request another font number.
700 to 710	Input an alternate font number. Refer to line 580.
720 to 750	Define character strings for use with the alternate font. These strings must be redefined with the Control-N character included. Control-N tells the LETTER routine to use the font specified by LAFONT. Note the two Control-Rubout characters in front of the string in line 720 and at the end of lines 720, 730, 740, and 750. These give carriage returns.
760 to 820	Adjust the character size, display the character strings, and end the program. Refer to lines 610 thru 640.

Appendix B

COMMAND SUMMARY

Function	Example	Purpose
ARC1	CALL "ARC1",30,30,90	Draws an arc centered on the point (30,30) and extending 90° counter-clockwise from the current cursor position (if SET DEGREES has been issued).
ARC2	CALL "ARC2",1;50,40,10,0,180	Draws an arc on device 1. The arc is centered on point (50,40) has a radius of 10, starts to the right of the center point, and extends 180° counterclock-wise around the point (if a SET DE-GREES command has been issued).
CIRCLE	CALL "CIRCLE",Z	Draws a circle around the current cursor position with a radius specified by Z.
CLDRAG	CALL "CLDRAG"	Clears the ENDRAG function.
CURSOR	CALL "CURSOR",X,Y	Enables the crosshairs, sets X and Y to the current cursor position, and enables the drag bit.
ENDRAG	CALL "ENDRAG"	Enables the current cursor object to be dragged with the thumbwheels.
LAFONT	CALL "LAFONT",5	Alternate print fonts may be selected. Control characters in the Character Display routine call would specify that Font 5 should be used.
LCENTR	CALL "LCENTR","Z"	Prints the letter Z centered on the present cursor position.

Function	Example	Purpose
LDIM	CALL "LDIM",10,10	Sets the size of the character envelope (10 user data units from character to character, and 10 from line to line).
LETEVN	CALL "LETEVN",C$	Displays the character string specified by C$, with narrow spaces for narrow characters.
LETTER	CALL "LETTER",1;A$	Prints the character string specified by A$ on device 1.
LHOME	CALL "LHOME"	Homes the cursor, taking the line size into account.
LMFONT	CALL "LMFONT",3	Characters are displayed in special print fonts. (Font 3 is specified.)
LMOVE	CALL "LMOVE",0,3	Moves the cursor three lines and no spaces up the display. Line and character size is set using other routines.
LQUAL	CALL "LQUAL",0	Sets character resolution either enhanced or normal. A setting of 0 calls for enhanced quality.
LRATIO	CALL "LRATIO",1.2,2	Sets the ratio between character size and envelope size (character width/character space= 1.2, line height/character height= 2).
LRESET	CALL "LRESET"	Resets all the Character and Symbol routines to initial or default values.
LROT	CALL "LROT",1	Rotates the character line. This example rotates the line to print at a 1 radian angle to horizontal.
LSCALE	CALL "LSCALE",1,0.5	Magnifies the character and envelope size. This example cuts the line and character height in half.

Function	Example	Purpose
LSIZE	CALL "LSIZE",11,8	Sets special width (11 user data units) and height (8 user data units) of all characters.
LSLANT	CALL "LSLANT",−1	Slants individual characters. This example slants characters one radian to the left.
LSMOTH	CALL "LSMOTH",0.8	Sets character smoothing. A 0.8 setting calls for relatively little smoothing. The LSMOTH routine is ignored if LQUAL is normal (1).
LVERTI	CALL "LVERTI",B$	Displays the character string represented by B$ on a vertical axis. Characters are displayed top to bottom.
SMOOTH	CALL "SMOOTH",0	Sets the degree of smoothness for arcs and circles. A zero provides maximum smoothing; a one is minimum smoothness, or an octagon.
STROKE	CALL "STROKE",1;Z1,X(1)	Draws a custom, user-defined character or symbol. The argument 1 is the device number, Z1 is a special command string, and X is an array of arguments for the commands given in Z1.
TYPEKY	CALL "TYPEKY"	Issues an automatic carriage return for use by a subsequent POINTER command. The POINTER command does not need an operator entry.

Appendix C

UNDERSTANDING ERRORS

The following error messages may occur while using the Character and Symbol ROM Pack. This list is supplied for convenience. A more complete list of error messages is given in the 4050 Series Graphic System Reference Manual.

Error Message	Meaning
12 INVALID COMMAND ARGUMENT IN IMMEDIATE LINE INVALID COMMAND ARGUMENT IN LINE xx	Improper syntax is used in the routine call. An invalid argument is given for the routine.
32 CALL NAME INVALID IN IMMEDIATE LINE CALL NAME INVALID IN LINE xx	The ROM pack is not installed or incorrectly installed. The ROM pack is defective.
36 UNDEFINED VARIABLE IN IMMEDIATE LINE UNDEFINED VARIABLE IN LINE xx	An input array or variable is not defined or contains undefined elements.
39 MEMORY IS FULL IN IMMEDIATE LINE MEMORY IS FULL IN LINE xx	There is not enough space left in the Random Access Memory for the routine to be completed. For some of the routines in this ROM pack, 0.75 K of available RAM is required. Use the MEM command in a preceding line, or restructure the program.
66 INVALID I/O OPERATION IN IMMEDIATE LINE INVALID I/O OPERATION IN LINE xx	The I/O operation or the device called is illegal or invalid.

INDEX

ARC1 routine 2-36
ARC2 routine 2-38
Arcs ..... 2-35—2-39
Character display routines ..... 2-26—2-33
Character envelope 2-3
Character transforms ..... 2-2 — 2-34
CIRCLE routine 2-35, 2-40
CLDRAG routine 2-50
Control characters 2-28
CURSOR routine 2-47
Default settings 2-34
Drag bit 2-47, 2-48
Ellipse 2-35
ENDRAG routine 2-48
Fonts ..... 2-12 — 2-16
INIT statement 2-34
Initial settings 2-34
Installation 1-4
LAFONT routine 2-18
LCENTR routine 2-29
LDIM routine 2-4, 2-6
LETEVN routine 2-33
LETTER routine 2-27
LHOME routine 2-25
Line feeds 2-24, 2-28
LMFONT routine 2-17
LMOVE routine 2-24
LQUAL routine 2-21
LRATIO routine 2-4, 2-7
LRESET routine 2-34
LROT routine 2-9, 2-10
LSCALE routine 2-8
LSIZE routine 2-4, 2-5
LSLANT routine 2-9, 2-11
LSMOTH routine 2-22, 2-44
LVERTI routine 2-31

Memory requirements C-1
Multiple segment rubber-banding 2-48
Overstriking 2-24, 2-30
POINTER statement 2-46, 2-47
PRINT statement 2-2, 2-27
Resolution ..... 2-20—2-22, 2-41
Rotation 2-9, 2-10
Slanting 2-9, 2-11
SMOOTH routine 2-41
STROKE routine 2-42
TYPEKEY routine 2-46

Tektronix
COMMITTED TO EXCELLENCE

MANUAL CHANGE INFORMATION

PRODUCT 4052R11 CHARACTER SYMBOL ROM PACK
CHANGE REFERENCE C1/481
MANUAL PART NO. 070-3794-00
DATE 4-7-81

EFF ALL SN

SCHEMATIC AND PARTS CHANGES

REMOVE:

U113	156-0140-00	MICROCRT,DI:HEX BUFFER

CHANGE TO READ:

U1	156-1598-00	MICROCRT,DGTL
U11	156-1598-00	MICRICRT,DGTL
U13	156-1598-00	MICROCRT,DGTL
U15	156-1598-00	MICROCRT,DGTL

Tektronix
COMMITTED TO EXCELLENCE

MANUAL CHANGE INFORMATION

PRODUCT 4052R11 CHARACTER AND SYMBOL ROM PACK

CHANGE REFERENCE C2/581

MANUAL PART NO. 070-3794-00

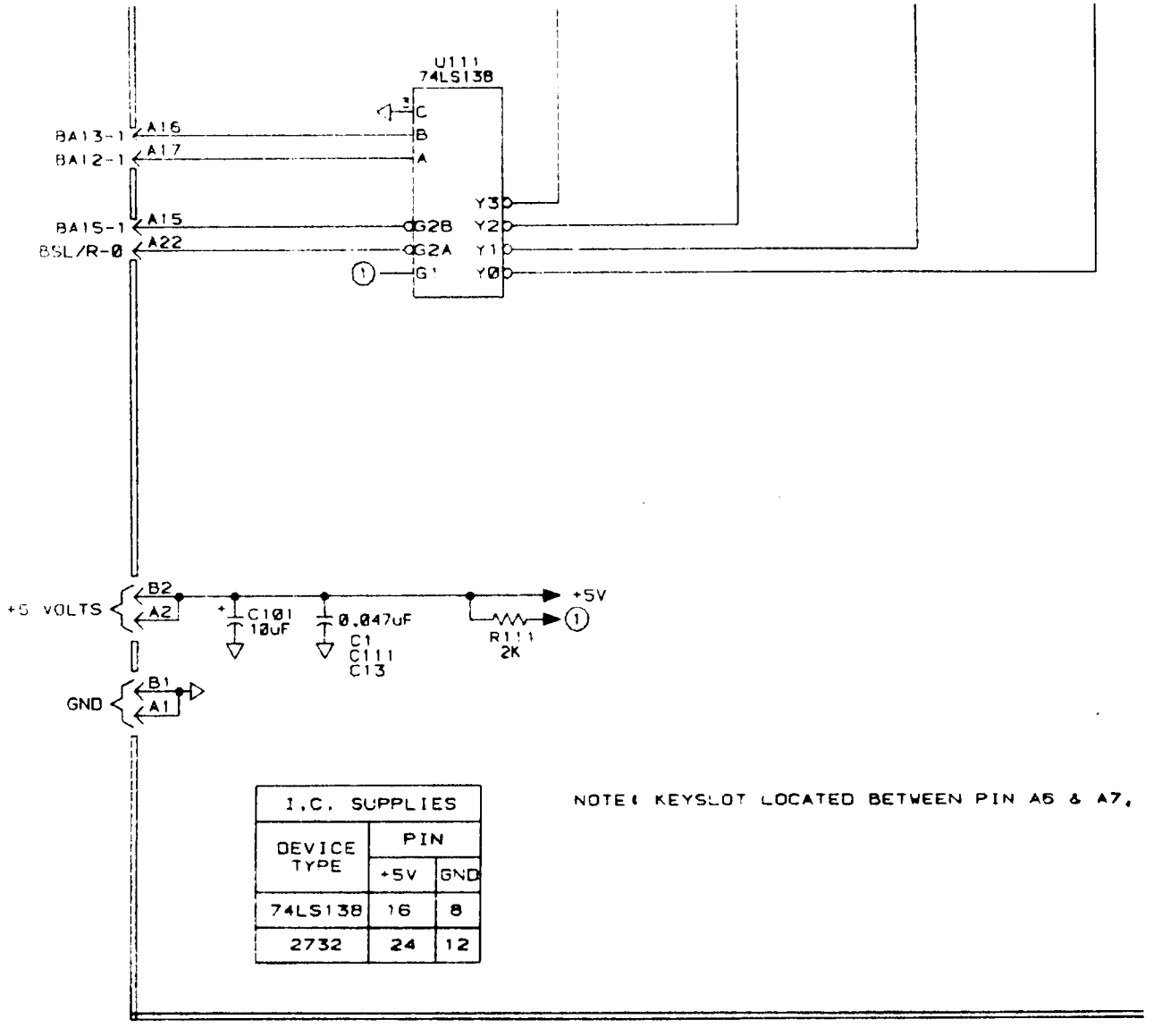
DATE 5-18-81

EFF ALL SN

SCHEMATIC CHANGE

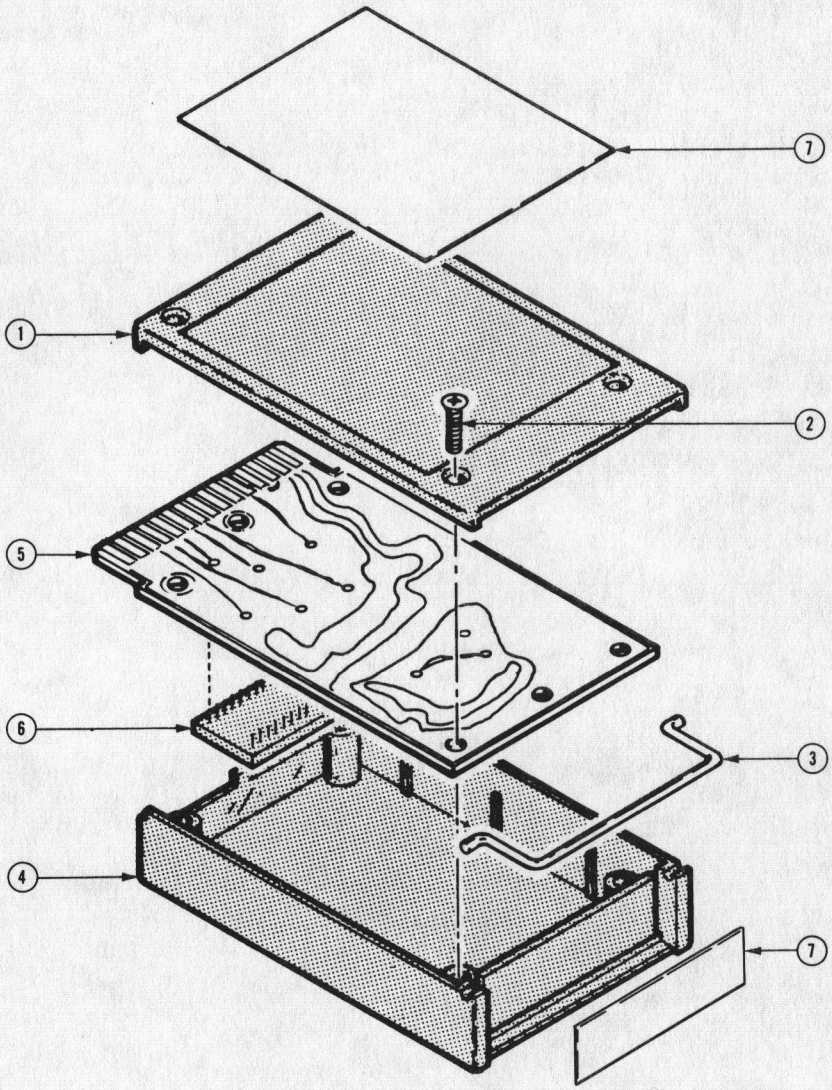
CHARACTER AND SYMBOL ROM PACK BOARD 670-7289-00

CHANGE AS INDICATED: Note that original inclusion of U113 was in error.



I.C. SUPPLIES		
DEVICE TYPE	PIN	
	+5V	GND
74LS138	16	8
2732	24	12

NOTE: KEYSLOT LOCATED BETWEEN PIN A6 & A7.


7
1
2
5
6
3
4
7

Section 4

DIAGRAMS

Symbols and Reference Designators

Electrical components shown on the diagrams are in the following units unless noted otherwise:

Capacitors = Values one or greater are in picofarads (pF).
Values less than one are in microfarads (μF).

Resistors = Ohms (Ω).

Graphic symbols and class designation letters are based on ANSI Standard Y32.2-1975.

Logic symbology is based on ANSI Y32.14-1973 in terms of positive logic. Logic symbols depict the logic function performed and may differ from the manufacturer's data.

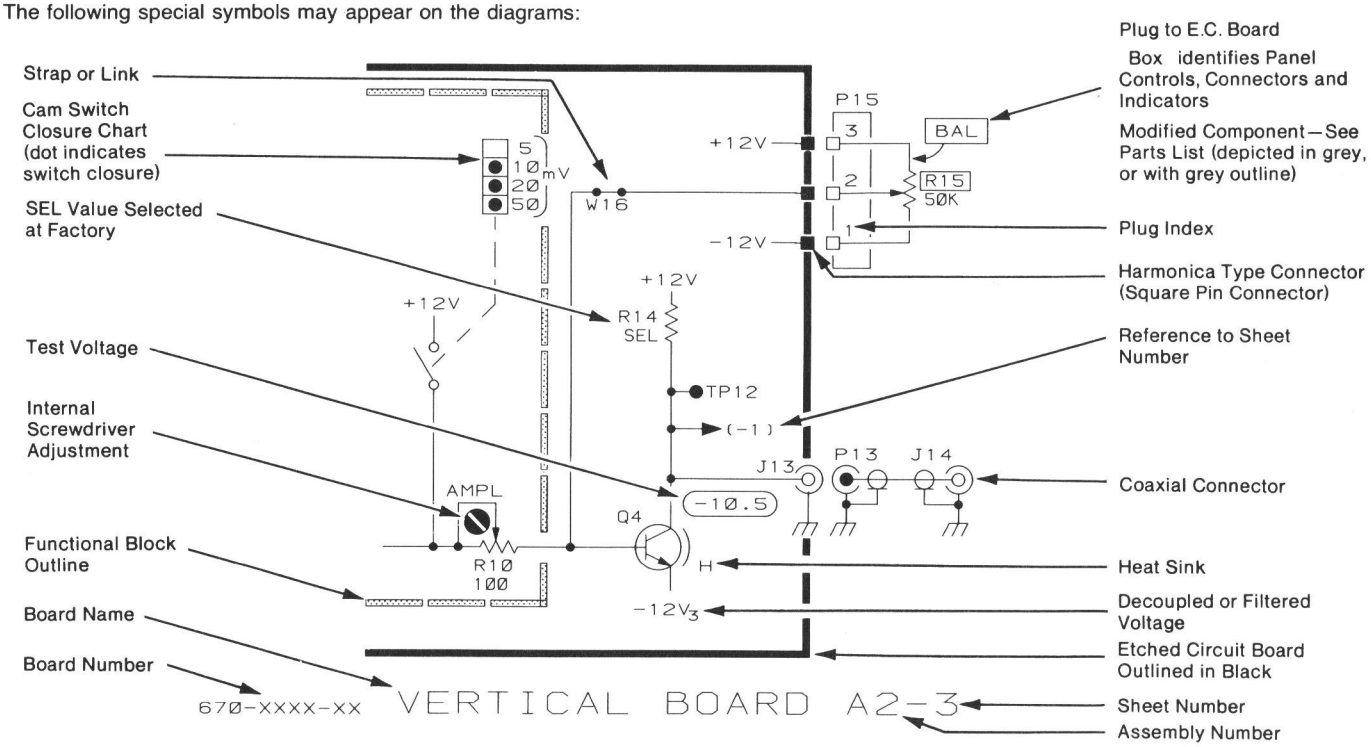
Abbreviations are based on ANSI Y1.1-1972. Other ANSI standards that are used in the preparation of diagrams by Tektronix, Inc., are:

Y14.15, 1966	Drafting Practices.
Y14.2, 1973	Line Conventions and Lettering.
Y10.5, 1968	Letter Symbols for Quantities Used in Electrical Science and Electrical Engineering.

The following prefix letters are used as reference designators to identify components or assemblies on the diagrams.

A	Assembly, separable or repairable (circuit board, etc.)
AT	Attenuator, fixed or variable
B	Motor
BT	Battery
C	Capacitor, fixed or variable
CB	Circuit breaker
CR	Diode, signal or rectifier
DL	Delay line
DS	Indicating device (lamp)
E	Spark Gap, Ferrite bead
F	Fuse
FL	Filter
H	Heat dissipating device (heat sink, heat radiator, etc.)
HR	Heater
HY	Hybrid circuit
J	Connector, stationary portion
K	Relay
L	Inductor, fixed or variable
M	Meter
P	Connector, movable portion
Q	Transistor or silicon-controlled rectifier
R	Resistor, fixed or variable
RT	Thermistor
S	Switch or contactor
T	Transformer
TC	Thermocouple
TP	Test point
U	Assembly, inseparable or non-repairable (integrated circuit, etc.)
V	Electron tube
VR	Voltage regulator (zener diode, etc.)
W	Wirestrap or cable
Y	Crystal
Z	Phase shifter

The following special symbols may appear on the diagrams:



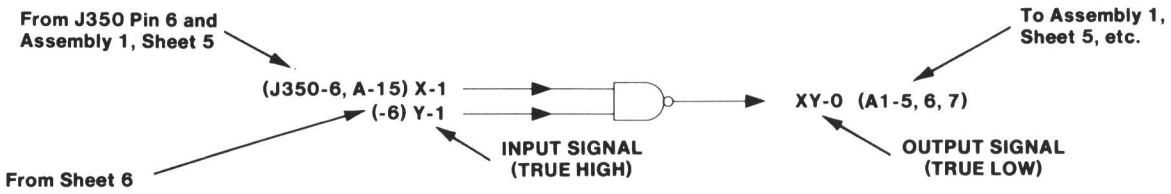
1. True High and True Low Signals

Signal names on the schematics are followed by -1 or a -0. A TRUE HIGH signal is indicated by -1, and a TRUE LOW signal is indicated by -0.

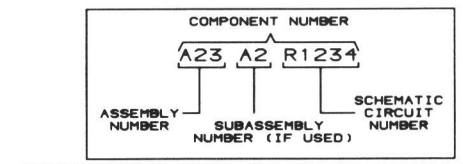
SIGNAL -1 = TRUE HIGH
SIGNAL -0 = TRUE LOW

2. Cross-References

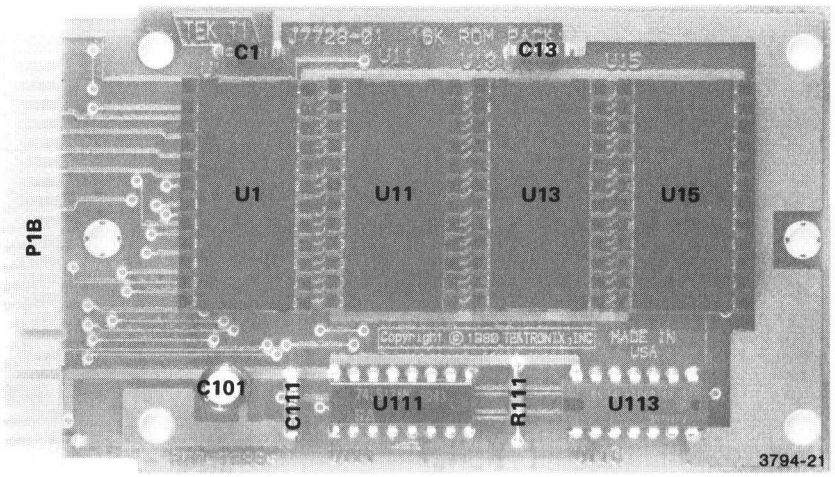
Schematic cross-references (from/to information) are included on the schematics. The "from" reference only indicates the signal "source," and the "to" reference lists all loads where the signal is used. All from/to information will be enclosed in parentheses.



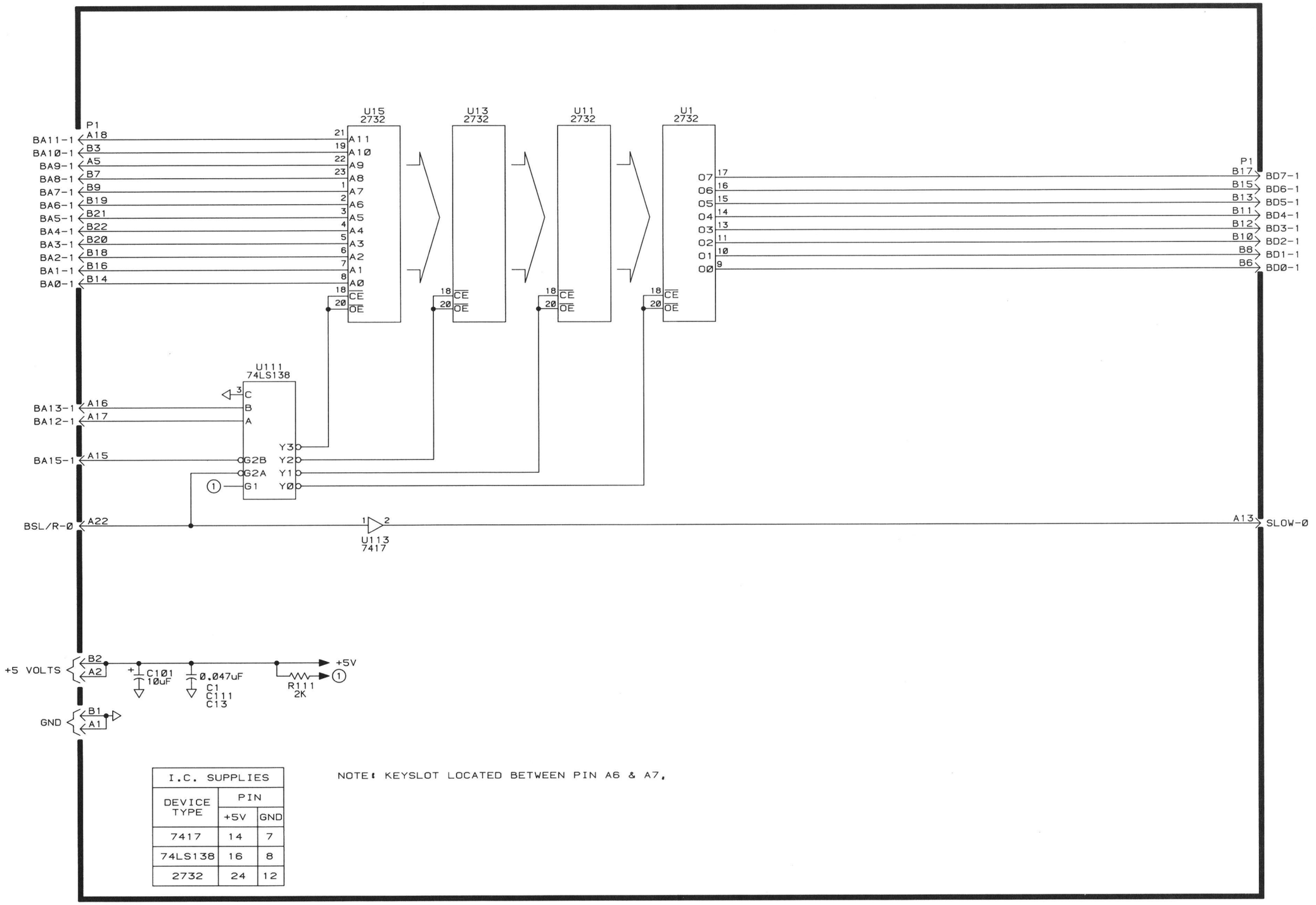
3. Component Number Example



CHASSIS-MOUNTED COMPONENTS HAVE NO ASSEMBLY NUMBER PREFIX— SEE END OF REPLACEABLE ELECTRICAL PARTS LIST.



ROM Pack (670-7289-00) Component Locations.



I.C. SUPPLIES		
DEVICE TYPE	PIN	
	+5V	GND
7417	14	7
74LS138	16	8
2732	24	12

NOTE: KEYSLOT LOCATED BETWEEN PIN A6 & A7.

INSTRUMENT:	NOTES	BOARD: 670-7289-00 16K ROM PACK	ASSEMBLY/SHEET
4052R11	@ 3794-1	CHARACTER AND SYMBOL ROM PACK BD.	A1-1 (1 OF 1)